众妙之门

THE SMASHING BOOK

网站UI设计之道（修订版）

[德]Smashing Magazine 著　贾云龙 王士强 译



人民邮电出版社
POSTS & TELECOM PRESS

数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

众妙之门

THE SMASHING BOOK

网站UI设计之道（修订版）

[德] Smashing Magazine著　贾云龙 王士强 译

人民邮电出版社

图书在版编目（ＣＩＰ）数据

众妙之门 : 网站UI设计之道 / 德国Smashing杂志著 ; 贾云龙, 王士强译. -- 2版（修订本）. -- 北京 : 人民邮电出版社, 2012.11
ISBN 978-7-115-29461-6

Ⅰ. ①众… Ⅱ. ①德… ②贾… ③王… Ⅲ. ①网站－设计 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2012)第219429号

版权声明

众妙之门——网站 UI 设计之道（修订版）

◆ 著 [德] Smashing Magazine
译 贾云龙 王士强
责任编辑 赵 轩

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京精彩雅恒印刷有限公司印刷

◆ 开本：880×1230 1/32
印张：8.5
字数：268 千字 2012 年 11 月第 2 版
印数：1– 3 500 册 2012 年 11 月北京第 1 次印刷
著作权合同登记号 图字：01-2010-3999 号

ISBN 978-7-115-29461-6

定价：39.00 元

读者服务热线：(010)67132692 印装质量热线：(010)67129223
反盗版热线：(010)67171154

内容提要

本书是知名设计网站 Smashing Magazine 上线以来的精华总结，来自全球的知名设计师无私地分享了他们多年积累的宝贵经验。全书共分 10 章，前 8 章讨论了和网站 UI 设计相关的八个不同的领域，覆盖面非常广，具有很强的操作性和专业性，全书逻辑严密、言简意赅，设计人员可以快速地找到自己想要的东西。第 9 章以访谈的形式介绍了多位专家对一系列重要问题的看法，入木三分，极富启发性。第 10 章以讲故事的方式，把 Smashing Magazine 的创办和发展过程娓娓道来，同时毫不保留地分享了两位创始人的成功秘诀和价值观，让人受益匪浅。

本书最大的价值在于它在传授具体经验的同时，也对设计思想和设计理念进行了深入的探讨，在“授人以鱼”的同时，不忘“授人以渔”。相信广大读者读完这本书，一定会有一种相识恨晚的感觉。

序　言

人可以使简单的事情变得复杂，通过常规的方式写一本书对我们而言算不上什么挑战，从撰稿者的名单可以证明这一点。我们希望尽可能多的人参与进来：作者、设计师和插画师。实际上，在圈儿内，我们把这个小小的项目称为“社区之书”。然而，这本书并不是这个社区成员的作品，而是由一个专业的作家团队创作的。在成书过程中（从头脑风暴到最终的版面设计），我们把读者也包含在了计划里面：我们在论坛（Smashing Forum）中向人们征集意见，并聆听他们的观点。当然，所有的这些前期的辛劳，也可以吊起读者的胃口。成果现在就握在你们的手中，这是 Smashing Magazine 的第一本书。我们希望至少做了一些大家看来正确的事情。

本书包含了 Smashing Magazine 精心准备、精心编写以及精心编辑的内容，这些都基于 Smashing Magazine 读者的建议和意愿。我们认为本书包含的主题是至关重要的，所以，我们希望本书的内容对您来说非常实用。在 http://smashing-links.com 提供的一份便捷的概述中，你会发现本书参考和提到的全部资源的列表。

请通过书评或评论的方式，让我们知道你对本书的看法。愿你们在本书中找到乐趣和酷毙了的体验！

Sven Lennartz（斯文·雷纳兹）

Vitaly Friedman（维塔利·弗里德曼）

目录

第 1 章

现代网络应用中的用户界面设计

Dmitry Fadeyev

什么是用户界面设计？什么是有效的用户界面设计？你打算如何把用户界面设计得更好？这一章，我们就来了解一下现代互联网应用程序中可视用户界面设计的理论演化和技术更新。

1.1　什么是用户界面（UI）？

“在你使用工具完成任务的过程中，你所做的操作以及工具的响应，这些结合起来就构成了界面。”—Jef Raskin

用户在第一次使用一个界面时，仍然能发现某些元素是他们所熟知的。

用户界面设计并不仅仅是考虑如何摆放按钮和菜单，更为重要的是考虑应用程序或设备如何与用户互动；更多时候，还要考虑产品与多个用户之间的互动。这就意味着，用户界面设计不仅要做出好看的产品外观，还要做出好用的产品。这不仅仅是调整一下按钮的位置、给他们选选颜色，而是要为完成一项任务而选择正确的工具。一个特殊的界面，是否需要使用按钮？如果确实需要，那么这些按钮将完成什么样的工作？这个按钮需要如何设计才能让用户了解这个应用程序的工作方式，并且轻松达到用户的目的。

在产品的开发周期中，尽早开始用户界面设计至关重要。正如 Jef Raskin 所说：“对消费者而言，界面就是产品[1]”。用户一直看到和使用的都是用户界面，而不是深藏在应用程序背后的技术体系结构。把界面做好，这将极大地提高用户使用产品时的愉悦感，这也能让其很容易就掌握产品的使用方法。所以，先从界面设计开始，然后再开始应用程序后台引擎的编码，要好于先开始编码，再给它包裹一层界面。

什么因素能成就一个伟大的用户界面?

在我们开始着手设计用户界面之前，首先应该理解什么样的用户界面才是出

[1] Raskin,J. 2000 The Humane Interface Addison Wesley。

色的用户界面，我们的设计要具备哪些品质才够得上“出色”。所有伟大的用户界面大都具有下面这 8 个品质或特点。

1. **清晰**——通过使用文字、流程图、层级图、图标等元素，避免用户对界面产生困惑。清晰的界面不需要手册，同时还能确保用户在使用的过程中减少犯错。
2. **简洁**——为了让界面清晰，可以通过在所有元素上添加浮动框说明或者标签这种简单的方法来实现，但这样做会导致界面膨胀臃肿。如果电脑屏幕上同时充斥了太多的东西，找到你想要的东西是非常困难的，用户使用起来就觉得困难乏味。如何让界面同时保持简洁和清晰，这正是设计伟大界面的挑战所在。
3. **熟悉**——人们总是对之前见过的东西有一种熟悉的感觉，有的界面即使是第一次使用，其中的有些元素我们还是会感觉很熟悉。在设计过程中，使用一些现实生活中公认的意象能够更好地帮助用户理解。例如，文件夹样式的选项卡通常在网站和应用程序中起导航作用，人们很快就能意识到它的导航作用，因为现实生活中，我们也是通过文件夹来分类资料的。
4. **响应性**——这意味着两件事情。首先，响应必须迅速，一个良好的界面不应该让人感觉反应迟缓。其次，界面应该提醒用户发生了什么事，用户的输入是否成功地处理，让用户了解这些信息反馈。并且明确告诉用户，他的请求已经被成功受理。
5. **一致性**——在整个应用程序中保持界面一致是非常重要的，这能够让用户识别出使用的模式。一旦用户学会了界面中某个部分的操作，他很快就能知道如何在其他地方或其他特性上进行操作，就好像他们早就知道似的。
6. **美学性**——尽管你不需要把一个界面做得非常有吸引力，但是有一个好看的界面会让用户工作起来更开心。况且，让用户开心绝对是一件好事。
7. **高效性**——时间就是金钱，一个伟大的界面应当通过快捷菜单或者良好的设计来帮助用户提高工作效率。毕竟这是科技带给我们的一个最大好处，它允许我们用更少的时间和努力来完成更多的工作。
8. **容错性**——每个人都会犯错，你的应用程序如何处理用户所犯的错误是对它的总体质量的最好测试。它是否容易撤消操作？是否容易恢复删除

的文件？一个好的用户界面不应当因为用户的错误而惩罚用户，而应该总是为用户提供犯错之后的补救方法。

想要设计一个包含所有特性的用户界面非常困难，因为各个特性之间总是相互影响。你在界面中增加的元素越多，用户在使用上所花的努力就越大。当然，反过来亦是如此。没有足够的帮助和支持，用户会茫然不知所措。如何设计既简洁、优雅，又清晰、一致的用户界面，是摆在用户界面设计师面前的最大难题。

1.2　可视界面设计的工具箱

可视界面设计就是让用户能够在他们的电子设备屏幕上看到你的界面的物理表现的过程。可视界面设计的主要目的是通过精心制作合适的可视界面，让用户理解应用程序的用途以及如何操作应用程序。

可视界面设计由一系列过程组成。选择合适的类型、对齐每个元素、将不同元素以一种有意义的方式组合起来，用户界面中不同的功能和特性借此传递给了用户。下面就是可视界面设计中的一些主要元素。

1.2.1　布局和定位

布局规定了界面中所有可视元素的结构。通过元素之间的距离，还确定了元素之间的层级和相互关系。元素距离比较近表明它们之间存在相互关联，例如图标下面的标签。定位也能改变整个界面的流畅性。例如将标签放在文本输入框的上面，而不是左边，这样我们就可以方便地从上往下扫描所有元素，而不需要先看左边，然后再找右边哪个输入框与其对应。

1.2.2　形状和尺寸

形状通常用来区分元素。例如，通过改变图标的外形，能够让它们被更快、更容易地辨认出来。尺寸经常用来表示重要性，越大的元素就越重要。尺寸也可以改变可点击控件的可用性。费茨定律告诉我们，一个可点击区域越大，用户将鼠标移动到该区域就更快。把经常使用的按钮做大，不仅让用户的点击变得更容易，同时提升了点击率。

1.2.3　颜色

很多地方都用到颜色。一种颜色在与它反差强烈的背景色下是能够吸引人的

注意（比如，在白色背景中的一个亮黄色警示箱）。不同的颜色可以表示不同的意义。例如，红色经常用来表示危险或停止（交通灯里的红灯），所以最好用红色来表示错误信息。而绿色一般用来表示成功或者继续进行的信号，所以应当用在表示这类信息的内容上。颜色也可以用来突出显示一些关系，例如在编辑的时候，根据按钮的关系，以不同颜色高亮显示，来帮助用户识别。

创造一个产品的外观和感觉并不是可视界面设计的主要目的，而仅仅是一个部分。最主要的目的还是沟通：通过沟通的行为帮助用户理解应用程序。

在使用颜色的时候要记住两件事。首先，不同文化中颜色代表的含义有所不同，所以确保你的目标用户能够识解你所选颜色的含义。其次，不要忘记色盲用户，在区分不同元素时，多花一些心思，使用一些颜色以外的技巧，例如一个图表中的长条。如果用户是色盲，他们就不能区分特定的颜色，多数色盲患者都是红绿色盲，所以你需要使用其他的方法，例如形状、材质等。

1.2.4　对比

一个元素周围的黑暗或者明亮将会对元素在界面上的可用性产生影响，这其实就是对比度在起作用。白色背景中的黑色文本具有高对比度，比起白色背景中的灰色文字，更容易辨认和阅读。调低一些元素的对比度，将它们融入到背景当中，用户就能够区分出重要元素和次要元素。

1.2.5　材质

在交互设计中有一个概念叫做 Affordance。Affordance 是一种能够让用户明白如何使用一种东西的特质。以把手为例，做一个单向门的最好办法就是在拉的这一面安装一个把手，在推的那一面装一个平坦的手柄挡板。看到门，人们自然就知道什么时候拉、什么时候推，因为门把手已经告诉了人们如何去使用。也就是说，提供了最好的互动方式，来让用户集中精力于正确的一个上面。

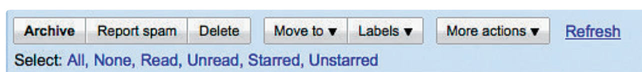
我们可以把这个主意应用到用户界面设计中，在屏幕上使用材质。例如，在可视界面中一些元素是可以拖拽的，就像在一个窗口的四角，我们可以对这个窗口进行缩放。为了让用户知道可以点击、拖拽它，通常会使用一些粗糙介质的突起。现实生活中，那些可以用手拉的物品通常会具有比较粗糙的材质。这个主意被转移到了屏幕上，不同的是我们不再用手指，而是用鼠标来进行拖拽。

1.3　制作有效用户界面的实用技巧

我们已经讨论过什么是用户界面，用户界面应该具备什么样的特点，以及我们可以使用的构建用户界面的工具，现在让我们看一些可以应用在你的网络应用或网站上的实用技术。

1.3.1　使用空白来构建联系

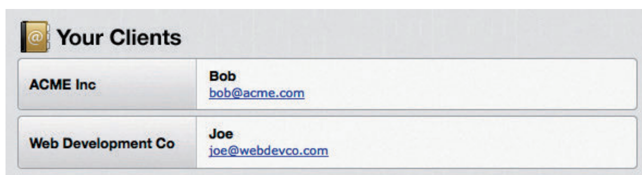
空白指在各种内容元素之间的空白区域，比如标题、正文和按钮。当在不同元素间建立联系时，空白是一个非常重要的工具。通过缩小元素之间的空白，能够把一些相关的项目组成一组。增加这些分组之间的空白能够更加强调这些分组与其他内容之间的联系。使用空白，我们可以分组相关的控件，也可以构建页面中元素之间的层级关系。



Gmail 的工具栏通过空白的分割突出了 3 个分组。每个分组中的按钮都对应着不同的功能。

1.3.2　使用圆角来定义边界

圆角通常用来改善图片元素的观感。它们看起来非常舒服，并且能够为你的界面增添平滑感，但这并不是圆角的全部功能。圆角通常定义一个对象的边界。当看到圆角时，你就知道已经到了容器的边界。如果是一个直角，后面再跟另一个直角的容器，它们之间的界限会不太明显。而圆角，或其他任何类似视觉上的处理，都能够让分界线变得更为明显。



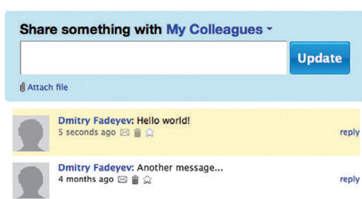
注意在 Ballpark 的客户列表中，圆角是如何来突出每条记录的界限的。在每个记录的中间也有两个直角，但是它们是用作相关数据之间的分割线，而不是一个条目的边缘。

1.3.3　用颜色来表达含义

颜色是一个重要的沟通工具，例如，定义不同的元素。你可以在应用程序中使用不同的颜色来区分不同功能的按钮。红色常被用在一些表示破坏性的按钮上，诸如删除或移动；蓝色一般用在标准按钮上；绿色则用在保存和更新操作按钮上。使用代码着色，我们可以分辨出屏幕中用户创建的各种各样的数据，让它们变得更容易阅读。

1.3.4　引导用户的注意力

使用动画来提醒用户关注。有时候，单独使用颜色和对比度还不足以吸引用户的注意。如果一件很重要的事情发生了，你也确实需要确保用户能够注意到，那么就使用动画。人类的眼睛非常适合捕捉移动的事物，特别是在固定背景中运动物体。如果用户在他的辅助工作的应用程序中添加一条待办事项，或者向购物车中加入一件商品，这个时候就可以使用动画来突出正在发生的事情。例如，当在屏幕中创建一个项目的时候，可以使用高亮的效果。对于那种使用很多 Ajax 技术的应用来说，这一点特别有用。在这些情况下，页面中的操作并不会导致页面的刷新，高亮的部分弹出来告诉用户发生了什么事情。



当你在 Yammer 上发布一条信息的时候，信息就会滑出并且高亮显示。

Latest project activity

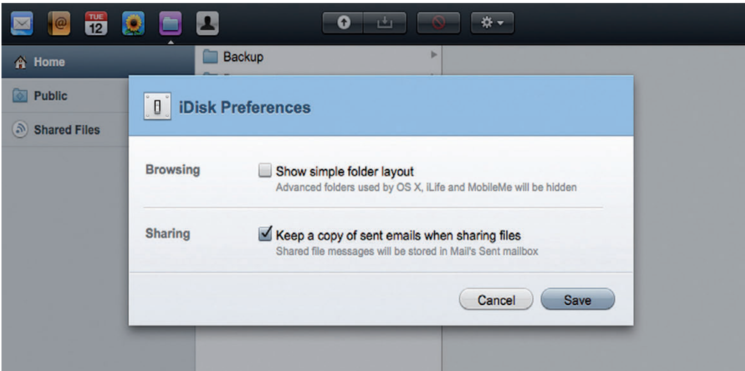
Task　Task closed – Make some coffee Change by dmitry_fadeyev, less than a minute ago
Task　Task created – Make some coffee Change by dmitry_fadeyev, less than a minute ago
Note　Note created – Hello world! Change by dmitry_fadeyev, 1 minute ago
Blog　Blog post created – New project created Change by dmitry_fadeyev, 1 minute ago
Project　Project created Change by dmitry_fadeyev, 1 minute ago

Goplan 通过在面板左侧放置不同颜色的标签，来帮助用户快速区分不同的项目，比如任务、笔记或者博客文章。

1.3.5　使用阴影

另外一个将用户注意力集中到某一区域的方法是使用阴影和加深背景。阴影

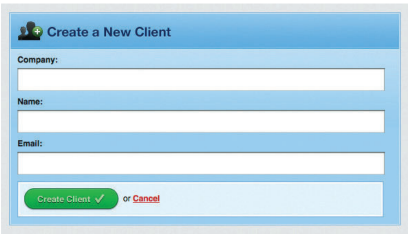
通常被设置在弹出菜单或者模式窗口四周，像一张毯子一样遮住了窗口四周的干扰元素。阴影降低了窗口下方其他元素的对比度，反过来就增强了用户使用区域的对比度。同样可以在窗口底部使用一个黑色（或亮一点）的半透明层，减少它所覆盖内容部分的视觉干扰，从而让用户的注意力集中在窗口上。



MobileMe 在模态窗口底部使用了黑色的遮罩层，同时在窗口四周使用了阴影。借此消除窗口以外元素的干扰，把用户注意力转移到窗口中来。

1.3.6 强调核心的动作

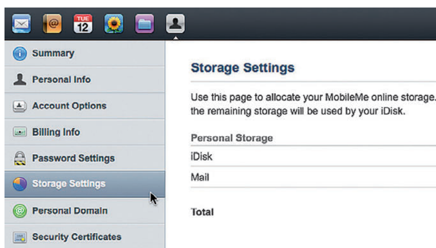
应用程序大多有自己特别的主要和次要界面。例如，当你在一个项目管理应用程序中创建项目时，主表单通常包含项目名称、项目截止日期、优先级，等等。在表单底部，你会看到一个“创建”按钮。通常你还会看到一个“取消”按钮或者文本的链接。因为用户不经常使用，所以“取消”按钮并不是很重要，因此，你就可以减少这个按钮的视觉“分量”。例如，可以把“创建”做成一个按钮，而“取消”只做一个没有任何视觉效果的超链接。这样中心就放在了主要的操作上，用户完成表单的时候就能很快地定位到操作按钮。



这是 Ballpart 客户创建的一个表单，注意看“Create Client”按钮与“Cancel”的区别是多么显著。

1.3.7 使用块状链接提高效率

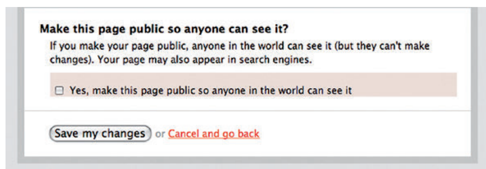
使用带填充的块状链接可以用来方便地将鼠标定位到链接。网络应用程序构建在 HTML 元素的基础上，也就意味会使用很多的锚（通常称为链接）标签。锚标签默认是“内联 (inline)”元素，意味着它的宽度和高度只是在文本的内部。反过来就是说，可以点击的区域和文本的范围一样，在很多情况下显得太小，用户不能够很精准、舒服地进行点击。而你可以对锚链接使用内边距来使它变大。对于列表中的链接，就像侧边栏，将锚链接转化为“块元素”是个更好的办法。通过使用 CSS 的“display”属性，我们可以转换一个元素的类型，所以给锚链接增加一个“block”属性，就能将其转化为一个块元素，块元素的宽度和高度就不再局限于文本的尺寸，而默认会扩展到包含它的容器的宽度。这种方法非常适合侧边栏中的链接。



MobileMe 侧边的导航面板，使用了带内边距的块状链接增大了可点击区域，用户能够快速地移动鼠标到这些区域，提升了可用性。

1.3.8 用动词作标签

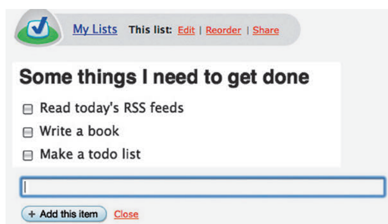
用户在使用应用程序的时候，总是准备去做一些事情，他们总是在想动词。“我想保存这个文件”或者“我想更新这篇文章”。在创建对话框或者其他需要用户处理的应用时，使用动词而不要使用像“是的”、“不是”、“好的”这样的表述。当用户看到像“好的”或者“取消”这样的选项时，他们需要去阅读额外的信息来搞明白他们要做什么。如果你使用动词，例如“保存”、“不保存”和“取消”，用户就不需要再看其他的消息，他们马上就知道这个对话框或选项是做什么的。



Backpack 在按钮或链接中很好地运用了动词，确保每个选择对用户都是明白无误的。

1.3.9 输入时自动 / 重新聚焦

在网页文件加载完成后自动激活主表单中的输入框会给基于表单的应用带来很大的好处。例如，Google 的搜索引擎页面自动将焦点定位在页面的输入框中，因为几乎所有到达 Google 首页的人都想在输入框中键入需要查找的内容。在其他的应用环境中自动激活输入框也非常有用，例如，在那种需要连续输入的应用中，就像创建购物清单，在输入第一个物品之后，用户想接着输入下一个，所以可以重新激活输入框，让用户快速、持续地进行输入。



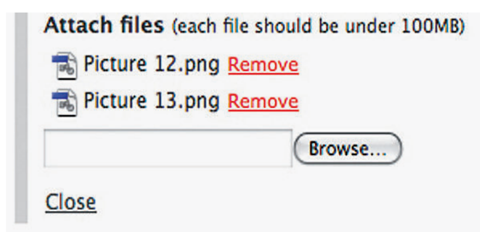
Tadalist 允许你快速、不断地输入待办事项。当输入一个任务并且单击回车后，程序会增加一个新条目，然后把焦点重新定位在输入框中，等待用户的下一次输入。

1.3.10 使用悬浮控件进行简化和解构

很多应用程序都有一套与内容紧密相关的控件，例如一个列表中的删除按钮和记录的编辑按钮。为了方便定位，通常这些控件放在了每条记录的后面，但是这种情况下就造成了大量的重复。多数的时间里，用户并不需要这些控件，需要时，也仅仅需要其中一个特定条目的控件。

如果使用悬浮控件，可以在用户将鼠标移动到一个特定区域后才显示，这样可以简化界面。例如，当鼠标停在你想编辑的记录上时，一个编辑按钮就会显现出来，但是其他记录的编辑按钮仍然保持隐藏。

悬浮控件是一个很棒的解构和简化界面的方法，但是在使用这个方法之前，你还需要考虑两件事情。首先考虑控件是否能被找到。用户在使用之前，往往需要先找到这个界面中的元素。你的控件是否做得足够显著以吸引用户注意？用户经

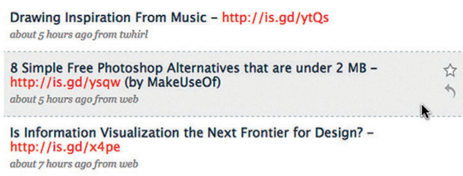


Basecamp 运行用户上传多个文件，但是每次只显示一个文件上传按钮。

常很自然地移动鼠标，停留在他们正在浏览的项目上，所以这也许并不是一个大问题，但是仍然值得去思考。其次，移动设备上的网页浏览器也许不能模拟鼠标的动作，那么这些控件将无法使用。

1.3.11　动态扩展表单

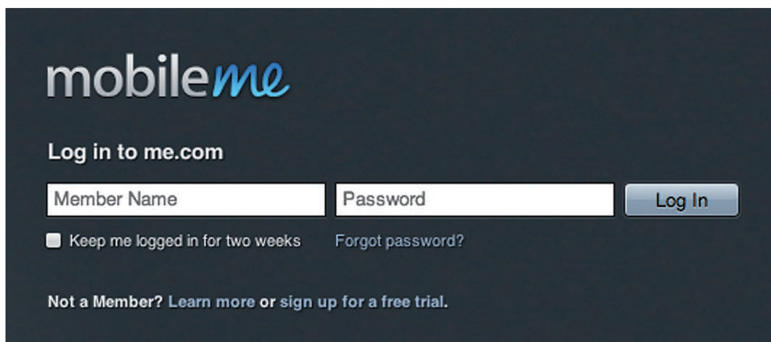
如果表单中的一个输入框中需要输入多个数据记录，例如，在一段文本之后附加多个文件，或者向数据库中的公司记录里添加多个人员信息。这种情况下，可以使用动态扩展表单，以一种优雅的方式来实现需求。因为用户每次只能在一个输入框中进行操作，所以每次也只需要显示一个输入框。当用户填完并且增加了一条记录后，程序会在新记录后再创建一个额外的输入框，允许用户键入更多的信息。这种方法替代了同时显示多个空输入框的形式，每次你只需要显示一个空的输入框，但是允许用户输入需要的内容。



Twitter 只有在鼠标停留在一个条目上的时候才显示“回复”和“收藏”按钮。

1.3.12　输入框中的标签

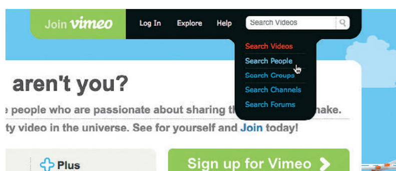
界面的简化需要不断地思考总结。如果你想要砍掉一些不必要的元素，并且为剩余的元素更好地利用空间，将标签放到输入框内部是一个巧妙的办法。这不仅节省了空间，也清楚地表明了与输入框对应的标签。



mobileMe 将标签放到输入框内部，使登录界面紧凑而有致。

1.3.13　上下文敏感的界面元素

如果希望为高级用户整合一些额外的功能，但是又不想让页面臃肿，这时就可以通过提供按需增加的上下文元素来实现。例如，你的应用中有一个带高级搜索的搜索条，在没有使用的时候只显示搜索条，当用户单击时再显示一些额外的控件。这样，界面仍然保持了简洁，实现高级的功能也只需要一次点击。



用户将鼠标移动到搜索条上时，会显示一个扩展的搜索条件，为高级用户提供一些额外的功能。

1.3.14　图标

使用图标能够简化界面、让界面看起来更有吸引力，但是在使用之前也有一些因素需要好好考虑。语言几乎总是比图标表达得更清楚。写一个标签非常容易，只要表明这个按钮的作用就行。然而在设计图标的时候，则需要用有效的意象来描述操作。如果图标是一些很本土化的东西，你选择的意象在不同的国家可能会有不同的理解（甚至连垃圾桶的形状在世界各地都是不同的）。

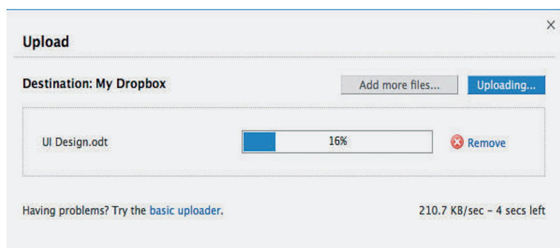
屏幕中同时没有太多的图标，每个图标都能够同其他图标明显地区分开，这样图标才能发挥出最好的作用。这意味着形状和颜色是创建一套成功图标的重要因素。使用正确的图标，能够让界面更容易使用。因为一旦用户熟悉了这些图标的含义，通过形状和颜色的区别，用户就能很快找到他们想要的图标。

1.3.15　使用加载指示器

没有人喜欢等待。等待意味着本来可以做一些有价值的事情的时间里，你却什么都没有做。但是很不幸，载入每个程序都需要花费一些时间，比如导出一个大文档或者通过查询返回结果，用户不得不等待。但是，可以通过使用加载指示器来让用户觉得等待的时间变短了[1]。在加载过程中，可以增加一个指示器，比如旋转的动画或者进度条。研究表明，在指示器存在的情况下，用户感觉等待时间变短了。加载指示器对短时的等待非常适用，但是对于一个需要一分钟甚至更多时间的情况呢？如果你在加载的这段时间里什么都

1　http://www.usabilitypost.com/2009/01/23/maing-wait-times-feel-shorter/。

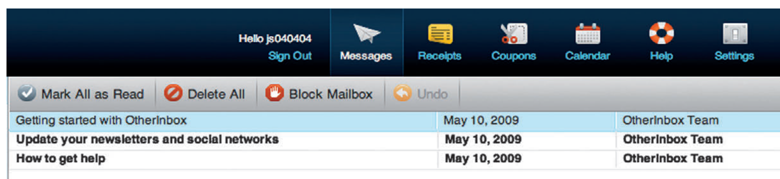
不能做（假设你已经为减少加载时间做了最大的努力），你还是可以做一些事情来改善用户的等待体验。利用这段时间提供娱乐或者信息。例如，很多应用程序在长时间的等待过程中会显示一些有趣的提示或技巧。如果用户确实做不了任何工作，起码他们可以学到一些新东西。



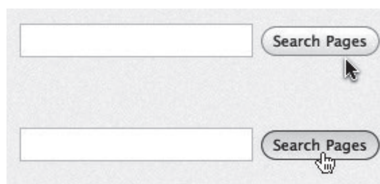
当用户通过 Web 界面上传文件时，Dropbox 显示一个进度条。对于类似的等待时间不定的事件，进度条是必不可少的，这样用户就不必猜测他们究竟还要等多长时间。

1.3.16　使用按钮状态提高响应性

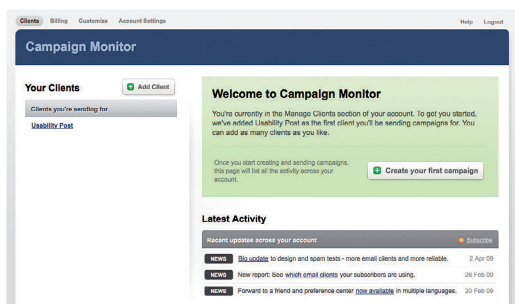
应用程序的响应性并不仅仅依赖于后台代码结构的优化。用户界面同样也扮演了非常重要的角色。在可视界面中被使用最多的一个元素就是按钮。在桌面应用程序中，按钮具有多种状态，最常见的是按钮默认状态和鼠标单击后的状态。



OtherInbox 在主导航中有一系列图标，图标底部都有相应的文字进行说明。



WordPress 中按钮的默认状态和按下的状态。



Campaign Monitor 在初次登录的时候提供了详细的帮助信息，同时有一个很大的按钮将你带到活动的创建页面。

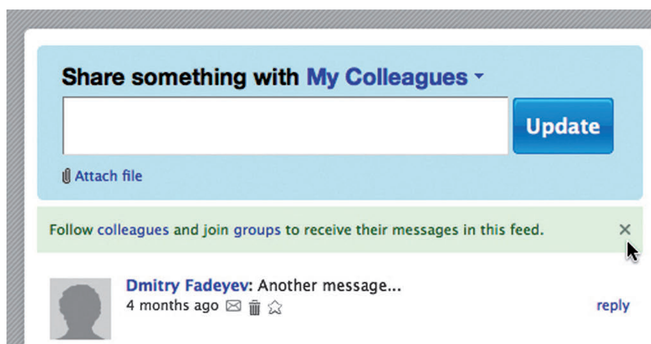
按钮按下的状态看起来就像真的被按下去那样。这个状态提示用户，他们的单击已经成功地执行了。像在桌面应用程序一样，按钮的按下状态同样可以使用在互联网应用中，为用户提供一些额外的反馈和响应。

1.3.17　帮助性的空白状态

当用户第一次加载应用的时候，屏幕上不会显示过多的内容；用户还没有输入数据，也没有什么好显示的。在这个空白状态中添加简短的帮助信息，来引导用户如何开始。为了让事情更加简单，你甚至可以在信息中加入一个链接，将用户带到推荐的操作。例如，一个管理电子邮件广告的程序，用户创建了一个新账户后登录，应该提供给用户一段帮助信息，其中包含一个链接到创建新广告的界面。

1.3.18　宣传应用的特点

用户对你的应用并不是十分了解，所以一些情况下在应用中合理使用广告是一个很好的办法。同时，由于互联网产品经常会进行一些细微的更新或者时不时推出一些新功能，所以通过广告的方式让你的用户知道这些变更就变得更有意义。一般情况下，可以通过在屏幕中的显著位置放置一个小块通知的方式来实现。通知区域应当吸引眼球但又不至于让用户分心，影响用户的正常工作。用户读了这个通知以后，应该可以关掉它。



在 follow 特性的主要输入区下面，Yammer 网站提供了一个非常有帮助的提示。用户读完以后，可以选择点击右边的关闭框关闭。

1.3.19　撤消操作

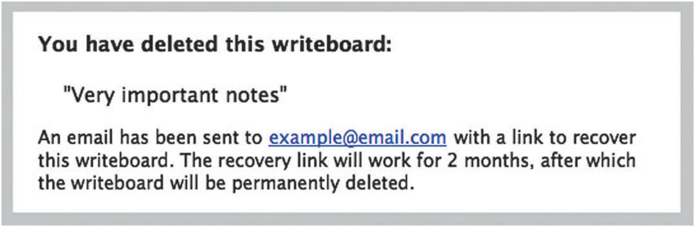
桌面应用程序最常用的特性之一是，值得信赖的撤消快捷键。不小心改变了正在编辑的文档图表格式？没问题，按 Ctrl+Z/Cmd+Z 组合键，应用程序会采取撤消操作，把你的文档恢复到最后一个动作之前的状态。撤消操作是容错界面的一个重要工具，它也可以应用在页面应用程序中。例如，你可以在网页应用程序中加入这些熟悉的快捷键或者显示一些带有撤消链接的临时通知信息。

The conversation has been moved to the Trash. Learn more Undo

每次在 Gmail 中删除一封邮件的时候，都会出现撤消操作的按钮，允许我们马上撤消删除操作。

1.3.20　还原操作

人们在删除一些东西之后，通常会反悔，所以在你的应用程序里增加额外的保护措施来处理一些比较重要的数据是一个明智的做法。例如，项目管理中的项目文件。用户删除一个项目时，将其进行存档而不是立即删除。如果用户改变主意，希望把他们需要的项目文件取回时，程序可以轻松地将它们还原。当用户发现所有的东西并没有丢失而感到非常高兴时，消耗一点用户的磁盘空间是值得的。


You have deleted this writeboard:

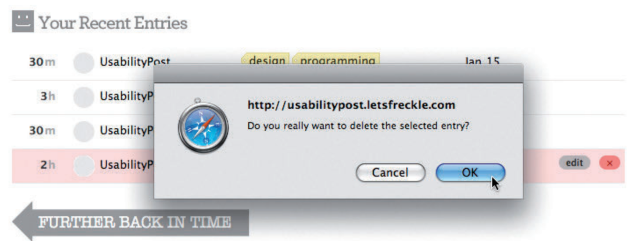
"Very important notes"

An email has been sent to example@email.com with a link to recover this writeboard. The recovery link will work for 2 months, after which the writeboard will be permanently deleted.

Writeboard 允许用户恢复两个月内被删除的文档。

1.3.21 确认对话框

当在应用程序中进行诸如删除记录这样的破坏性操作时，提供一个确认对话框，提醒用户是否确实要进行接下来的操作是非常必要的。特别是当删除按钮的位置和其他控件非常近的时候。如果用户不小心点击了删除按钮，在生效之前用户可以取消操作。但是，不要过度使用这个特性，因为不必要对话框拉长了操作过程并让人生厌。



Freckle 使用确认对话框来确保用户的误操作不会造成更大的损失。

1.4 结论

日本哲学家 Kaizen（中译音为“改善”）强调，应该通过微小、逐渐的过程来获得持续的改进和提高。用户界面设计，特别是现代网络应用的用户界面设计，并不需要一个完成的状态，因为我们总是不断地对其进行改善和升级。记住，你也应该像 Kaizen 一样思考。

传统的 CD 的分发模式有一个很大的缺点，即一旦产品烧录完成并寄送出去后，你就无法改变它……或者，不是那么容易改变 。你可以发布一个补丁，

但是用户必须下载并安装它才能生效，你无法确保每个人都会进行更新。另外，为每一个细微的改变发布补丁是不可行的，所以你必须在第一次发布的时候尽量做到完美，而这就需要在发布之前进行大量的测试。SAAS（软件即服务）的模式改变了一切。通过网络运行你的网络应用的好处是，你可以随时进行微小的更改，而用户马上就能够实现这些改进。

这就意味着，现在不必在第一次发布之前将界面做到 100% 正确。如果一些东西不合适，你可以修改它。通过观察用户与界面之间的互动，你就能找到让界面工作得更好的感觉，也能发现使用时的冲突。不管是哪一部分的用户界面需要改进，你可以很容易地为所有用户进行更新。这就是 Kaizen 的界面设计方法：微小的、逐渐的、常规的改进。你不再需要担心必须在第一次就将界面做到完美。相反，通过不断地改进，你将很快获得从实际使用中进化而来的伟大界面。

第 2 章

CSS 布局的艺术和科学

Jacob Gube　Kayla Knight

在现代网页设计中，网站布局设计工作需要耐心、细致以及扎实的 CSS 知识。设计元素决定了整体设计的视觉流向和层次结构，同时页面布局构建了整个网站的骨架，为页面中的元素提供了呼吸和生存的空间结构。但是设计网页布局确实非常复杂耗时，设计师的耐心在面对各种浏览器之间的不兼容以及多种风格和样式之间的利弊权衡过程中一点点被侵蚀。

好消息：已经有一些实用的指导可以帮助我们有效、可控地完成页面布局。本篇文章将会介绍多种类型的布局，并且分别介绍它们的优点和缺点，以及在何种情况下使用最好。同时，为了从总体上更好地理解 CSS 布局，我们还会涉及一些主要技术和相关话题。

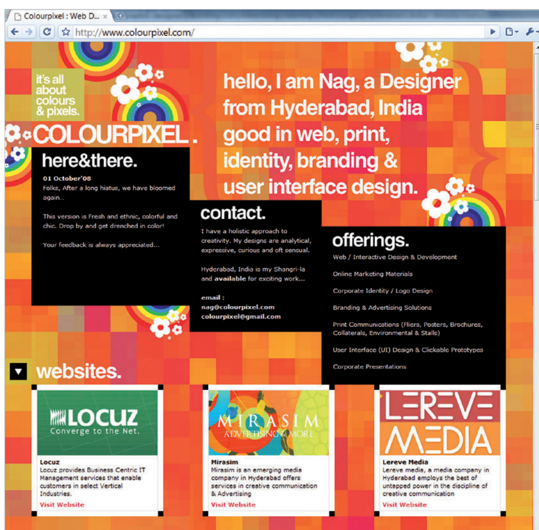
探寻建立一个完美、普适布局的过程最后都归结为对现存的各种方法的比较，这个过程还会极大地受到设计师的技能，以及在像素级、自适应、灵活性布局间选择的影响。基本上，设计师有 4 种可选择的布局类型：固定宽度布局、自适应布局、弹性布局和混合布局。这些布局的名字暗示了它们是否能够根据用户浏览器的尺寸调查自身的尺寸。

2.1 固定宽度布局

固定宽度的布局，就像它的名字一样，这是一种以像素为单位，宽度固定的静态布局。这类布局的外层宽度都是固定的，内部使用了百分比或者固定的宽度。设计师对内容区域周围的设计元素就有了更大的控制力，并且能够更精确地设计内容和导航的宽度。设计师根据自己的经验和探索，可以对页面中的字体、图片和平衡关系做更精确的规划。固定宽度的设计可以确保所有设计中的平衡关系能够始终如一，不管用户使用什么样的浏览器，这种平衡永远不会被打破。

固定宽度布局正是由于其简洁、直观、易于实现和定制的特性，成为了一种非常流行的选择。使用固定宽度布局，我们可以确保在不同分辨率下设计效果保持一致。我们不用去猜测用户使用什么客户端，测试也变得简单，从而总体上缩短了开发时间。特别是固定宽度布局使用绝对的像素值，这比相对

单位的百分比或者 em（基于字体大小）更能让设计师感到舒服和直观。严格按照像素来定位内容区域也许就是固定宽度布局最为显著的特点。



Colourpixel 的首页就使用了 950 像素的固定宽度布局。上图就是用户将浏览器设置为 1045 像素宽度时的效果。

固定宽度布局所用的宽度往往由显示器的平均分辨率决定。当下，大多数用户都已经拥有了 1024×768 或更高分辨率的显示器，所以大多数设计师选择了使用 960 或者 1050 像素（px）作为固定宽度布局的宽度。这个宽度可以避免水平滚动条的出现，并且为多数浏览器中都会出现的 20 像素的垂直滚动条留下了充足的空间。但是，对于低分辨率的屏幕，我们需要多加注意，如果固定宽度布局的宽度过宽，则会出现水平的滚动条。

固定宽度布局的一个优势在于设计要素易于定位，结果容易预测。确定了布局的宽度，定位元素就不会很复杂。另外，即使一个网站设计师考虑兼容最小的 800×600 像素分辨率显示器，在高分辨率的情况下，内容区域仍会清晰可辨。那些从平面设计领域转行到基于网页的设计领域的设计师趋向于选择固定宽度布局，因为他们热衷于精确的控制，毕竟在传统的印刷设计中普遍采用绝对定位的方式。



同一个网站在不同分辨率下的情况：浏览器的分辨率为 1680 像素，这时布局的右侧显示了大量的空白。

固定宽度布局的缺点是没有充分利用用户显示器的全部显示区域。当然，屏幕的分辨率不一定会和浏览器的分辨率相同，因为很多“宽屏”不会最大化浏览器窗口[1]。但是，如果用户有一个很大的宽屏显示器（例如，分辨率 1900×1200 像素，浏览器分辨率 1500 像素），屏幕中很大一部分没有被使用的假设就比较合理了。例如，浏览一个 750 像素宽的固定布局，空白区域的水平宽度会和布局一样宽。

因此，固定宽度布局产生了过度的空白，扰乱了黄金比例、三分构图、总体平衡或者其他的设计原则。那些顺畅的纹理、图案和连续的图片也需要适应高分辨率的情况。最后，强烈推荐使用一个居中的 DIV 来维护固定宽度设计中的平衡感。

```
#wrapper { margin:0 auto; }
```

这段代码可以实现这个效果。如果不这么做，在那些拥有高分辨率显示器的用户看来，整个布局会藏在左上的角落里。

那么，如何确定什么尺寸是最合适的呢？许多用户有了很大的宽屏显示器，

[1] 高分辨率的显示器带来了屏幕分辨率的提升，但是用户趋于避免使用全屏模式。在很多情况下，用户的浏览器保持了各种各样的尺寸，同时旁边还运行着其他的应用程序（比如Twitter应用、Rss阅读器或者即时通信工具）。参见“50.4% of respondents maximize windows”（456bereastreet.com）。

但是更多的人还在使用普通尺寸的显示器，还有一些仍在使用低分辨率显示器。不管是哪种情况，多数用户都会将窗口调整到他们喜欢的尺寸，而不是使用最大化的、标准的或者和屏幕分辨率成比例的窗口大小。

在实践中，为网站固定宽度布局找一个最佳的平衡点通常需要对网站的用户资料进行深入的研究。通常的做法是评定访问者的浏览偏好，然后相应地调整设计。最常用的方法是分析网站的统计数据，找到使用最多的分辨率，并分析有多少用户使用较低的分辨率。根据通常的经验，可以将宽度设置在800~1000 像素之间。作为专业人士，我们必须为用户而不是我们自己创建合适的布局，即使我们的设计在高分辨率的宽屏笔记本上看起来很糟糕。

下面我们就来创建一个简单的两栏的固定宽度布局。这个布局有一个放置Logo 的标题区域、主导航以及一个实现额外信息的底部区域。HTML 代码如下所示。

```
<div id="container">
    <div id="header">
      <!-- logo, navigation -->
    </div>
      <div id="content">
            <!-- main content -->
</div>
<div id="sidebar">
       <!-- sidebar -->
</div>
<div id="footer">
      <!-- site information -->
     </div>
</div>
```

为了固定宽度，在 #container 层上设置 width（宽度）属性，同时为左右两栏设置固定宽度。为了让头部和底部与容器等宽，设置宽度为 100%（尽管

在大多数的时候这个属性是不必要的，但这是一个很好的习惯，可以确保这两个区域确实与容器等宽)。



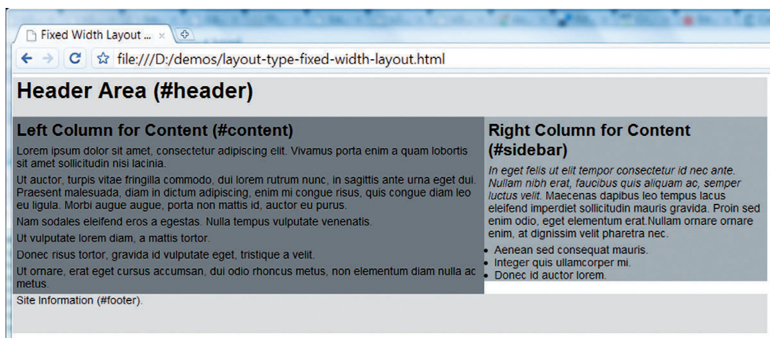
24ways 网站使用了固定宽度布局。#content 层的宽度固定为 900 像素。

本例中使用了 float 属性来让左侧和右侧的栏目正确显示 (当然，我们也可以使用相对或者绝对定位)，同时使用 clear 属性确保底部显示在两个栏目下方。另外，你还需要在其他类型的浏览器上进行测试，确保布局能够在不同的浏览器上正常显示。

```
#container {
margin: 0 auto; /* Centering the layout horizontally */
width: 960px;
}
#header {
width: 100%;
}
#content {
width: 600px;
float: left;
display: inline; /* IE 5/6 Doubled Float-Margin Bug 消除 IE5/6 中的浮动元素边距加倍的 bug  */
}
#sidebar {
width: 330px;
/* 960px - 600px - 330px = 30px is used to separate columns */
float: right;
display: inline; /* IE 5/6 Doubled Float-Margin Bug */
```

```
}
#footer {
width: 100%;
clear: both;
}
```

选择固定宽度布局还是其他类型的布局，其实是一个设计师愿意给用户多少自由的问题。前者把设计的决策交给了设计师而不是用户；后者则允许用户根据喜好调整字体、图片以及列宽。



一个固定宽度布局的例子。

对于固定宽度布局，人们普遍认为，从长远看来，其代码维护困难。如果设计中使用了固定尺寸的字体（尽管 IE6 的用户完全不能改变文字的尺寸），有一天客户要求加大某些特定的设计元素的尺寸（例如文本），重新计算并修改文本的尺寸将会成为一场“噩梦”。一个比较好的方法应当是使用元素之间的比例关系来决定设计的尺寸（例如使用相对值），这就是接下来要介绍的灵活布局（自适应布局和弹性布局）。

2.2　自适应布局

具有灵活变动宽度特性的布局，也被称为流式布局，这种布局会根据网页浏览器的窗口大小相应地调整自己的宽度。在这类布局中，容器中的主体部分的宽度以当前浏览器的窗口宽度为基准，按照一定的百分比进行定义。当浏

览器的尺寸发生变化时，布局也发生变化，因为用户调整窗口的操作已经改变了布局所依赖的基准。换句话说，这类布局中各区域的宽度会随着用户浏览器窗口的调整自动变化。所以，如果用户觉得正在浏览的文章的行宽不是很合适，通过简单地调整窗口尺寸，内容区域就会自动进行调整。

这类布局中，布局元素之间的关系至关重要，因为布局中子元素的宽度是以父元素的百分比设置的（不是相邻的元素）。例如，如果布局容器的宽度设置为 90%（他的父元素为浏览器窗口），导航区的宽度设置为 80%，那么导航区的宽度就是浏览器宽度的 72%（1×0.9×0.8=0.72）。由此可见，设计流式布局需要对设计的结构有深度的了解。对设计师而言，过程也许痛苦，但是成功的设计会让网站更加灵活地面对使用不同终端的用户[1]。

流式布局能够尽可能多地使用浏览器的可用空间，设计师可以有足够的空间来平均排布页面中的内容。设计得当的布局可以保持空白区域始终一致，不管浏览器或者屏幕的分辨率如何变化，这样的设计显得更加一致并且对用户友好。同时，这类布局也消除了低分辨率显示器上刺眼的水平滚动条。

和固定宽度布局不同，流式布局赋予用户更大的自由来调整布局的宽度。

和固定宽度布局不同，流式布局赋予用户更大的自由来调整布局的宽度。伴随着这种自由，设计师将严格、一致的像素布局转移到一种更加灵活（也因此更有风险）的环境中来。这时对网站中特定元素之间比例关系的定义变得相当抽象和困难。这也正是多数设计师避免使用流式布局的原因，这种布局剥夺了设计师在像素级精度上预测定位元素的能力。

设计师在用相对值替代绝对值的过程中，经常将这些值进位或者去位（例如，将 15.43333% 变成 15.43% 或者 15%），这样设计就变得更加不精确 。最后的设计也许在一些低分辨率屏幕上表现良好，但是在高分辨率情况下会发生奇怪的现象，诸如布局中显著的空白、内容区太宽难以阅读等。

事实上，内容区过宽的结果是流式布局最显著的缺点。由于内容区的宽度取决于用户视窗的大小，宽屏的用户会发现每一行都是如此的长，以至于阅读起来很不舒服。如果一个使用流式布局的页面中没有这么多的文字，用户将看到孤零零的一行字，侧边栏高得非常不协调，周边还充斥着大量的空白。

---

[1] 当然，很多终端用户并不知道他们可以通过调整浏览器尺寸来调整行宽，但这是另外一个话题了。

为了获得最佳的阅读效果（每行 47 到 86 个英文字母），设计师通常为内容区 #content-block [1]设置在此基础上加 10% 的宽度。除非设计的可读性会在低分辨率情况下变得很糟糕，一般不会设置最小宽度。

在 CSS 中，使用 CSS 属性“max-width”来定义布局的最大宽度。几乎所有的主流浏览器都支持这个属性（IE 7+、Firefox 1+、Safari 2+、Opera 9.2+），当然 IE 除外。

为解决这个问题，设计师通常会在内容区域设置一个最大宽度，以此来将布局限制到一个较为理想的宽度。

但是，仍有一个变通的解决方法。注意，在 IE8.0 版本中，IE 不再支持微软的动态属性（MSDN IE Blog“Ending Expressions”），即 CSS 表达式。所以 IE6 中的错误修正并不会在更高版本的 IE 中产生副作用。动态样式属性允许我们使用类似 JavaScript 中的结构控制和数学计算的语法，网页开发人员 Svend Tofte 在一篇如何解决 IE 浏览器中最大宽度问题的文章中有所讨论。根据文章中的示例，我们可以限制 #content-block 的宽度（svendtofte.com）。

```
#content {
max-width: 30em;
width: expression(
    document.body.clientWidth > (500/12) *
        parseInt(document.body.currentStyle.fontSize)?
      "30em":
      "auto" );
}
```

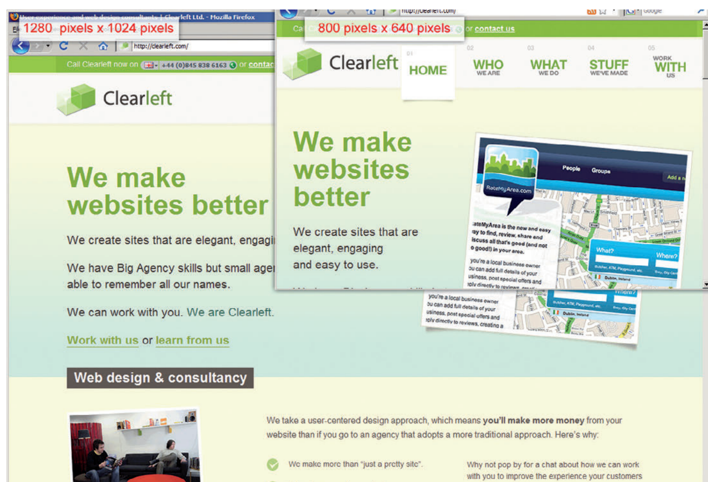
代码检查用户相对浏览器宽度的字体尺寸设置（以 pt 为单位）。Tofte 认为，不管在什么情况下，使用默认 12pt 的字体时，500px（像素）要比 30em 宽，所以上面的代码中使用了 30em 作为布局的宽度。

[1] 当然，这也取决于设计师为容器选择合适的宽度值或者使用特殊数值来计算布局元素宽度的能力。根据W3C的规范，在CSS中使用实际数值（例如82.345%）是可以的，但是不能确保在所有浏览器中被取整，这也是为什么大家更喜欢整数值的原因。

其原理是，如果内容部分的宽度大于 500 像素与 12 的比值乘以用户字体的设置，就将宽度设置为 30em。相反的情况下，让宽度保持自动(使用了“auto”属性)。

由于动态样式属性没有包含在 W3C 的 CSS2 规范中，使用这种方法会让样式表无法通过验证。读者可以根据自身情况使用，能够为市场上仍然是最流行的浏览器提供带有最小宽度和最大宽度的流式布局，付出这点代价也许是微不足道的。

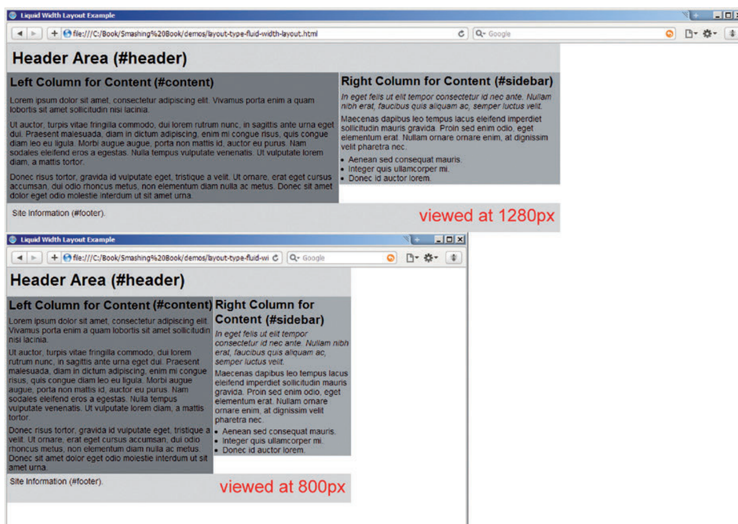
Clearleft 网站就是使用了自适应宽度布局的例子。下面两幅图展示了网站在 1280 和 800 像素情况下的表现。注意看内容区域是如何自动调整“We make websites better”这句话的，在 1280 像素时的两行变成了 800 像素时的三行。



最后，我们使用上面固定宽度布局的 HTML 结构来实现一个简单、纯粹的两栏自适应布局。再次提醒，记得使用多种浏览器来验证结果是否跨浏览器兼容。

```
#container {
  margin: 0 auto;
  width: 75%;
}
#header {
  width: 100%;
```

```
}
#content {
width: 60%;
float: left;
display: inline; /* IE 5/6 Doubled Float-Margin Bug */
}
#sidebar {
  width: 40%;
  float: right;
  display: inline; /* IE 5/6 Doubled Float-Margin Bug */
}
#footer {
  width:100%;
  clear:both;
}
```



上图为一个自适应布局的示例，容器的宽度设置为窗口的 75%，并且分别在 1280 和 800 像素的环境下进行了测试。

在为自适应布局选择一个合适的宽度时，使用“width：auto”或者其他的百分比值会非常合理，这样确保了布局自动占据浏览器的可用空间。对于最小宽度，可以使用像素值，例如 760（像素）。这个限制确保了所有自适应布局即使在最低的分辨率情况下仍保持可读。至于最大宽度，通常的做法是使用 em 值（例如 90em），这样就能在文本宽度不失控的情况下，保持不同分辨率情况下的稳定。让人高兴的是，如果用户调整浏览器的字体设置，最大宽度会发生相应的变化。

一个经典的计算这个比例的方法是，通过将给定像素高度元素与容器的尺寸相除得到的字体大小（即上下文）。

在实现自适应布局时，我们使用了相对值来表示外边距和内边距（当时，是相对于视口的宽度），这样当窗口尺寸变化时，外边距和内边距同样发生变化，同时保持了比例和尺寸一直灵活。这种技术被称为“concertina padding”，由 Richard Rutter 提出（“Web Standards Group：Ten Questions for Richard Rutter”，webstandardsgroup.org）。使用这种技术后，当页面尺寸改变时，外边距和内边距同样发生变化，同时会检查内容区以及文本行的长度。

2.3　动态网格布局

网格设计体现了秩序和精确。本质上，网格允许使用对齐、对称、适当的空白，从战略和系统上来组织排版和设计元素。最终实现令人愉悦和易读的设计。网格布局可以在多种布局类型基础上实现，但是设计师们通常认为弹性的布局要难得多，无法与传统的网格体系相提并论。但是本章中，这并不是必要的。

本质上，通过合理地使用 DIV 层、百分比和一些简单的数学运算就可以创建一个自适应网格布局。这个想法最早是由 Ethan Marcotte（自适应网格，alistapart.com）提出的，他认识到“如果我们不把字体尺寸仅仅认为是像素，而是测量它们容器的一个属性，我们就可以将网格布局中所有的元素都这样来测量。”他的基本意思就是使用相对单位、百分比和一个简单的除法运算，来找到替代固定布局设计中像素宽度的方法。因为 em 单位只是与内容有关，布局中的其他元素也可以这样对待。Ethan 指出，“这种方法允许将基于像素的固定宽度布局转换为基于百分比的布局，同时在缩放时保持各个网格之间

的关系。”

网格秩序体现了秩序与精确。本质上，网络允许使用对齐、对称、适当的留白，从策略和系统上来组织排版和设计元素。

一个经典的计算这个比例关系的方法是，通过将给定像素高度元素与容器的尺寸相除得到字体大小(即上下文)。最后，我们得到了期望的、使用相对值的、类似 em 或百分比的比例关系。这个公式就是：目标 ÷ 环境 = 结果。

如果我们将这个公式应用到设计中，就可以得到一个稳定的自适应布局。但是，我们实际上是希望得到网格布局，所以接下来我们要创建一个总宽为 988 像素，等分为 7 个 124 像素宽、之间有 20 像素分割的固定宽度布局。

假设现在布局中有一个 700 像素宽的 <h1> 元素（横跨了 5 列和 4 个空槽：5×124 像素 +4×20 像素 =700 像素）。这个元素应该为块状元素并且在布局中居中（也就是具有 288 像素的外边距，左右各 144 像素）。

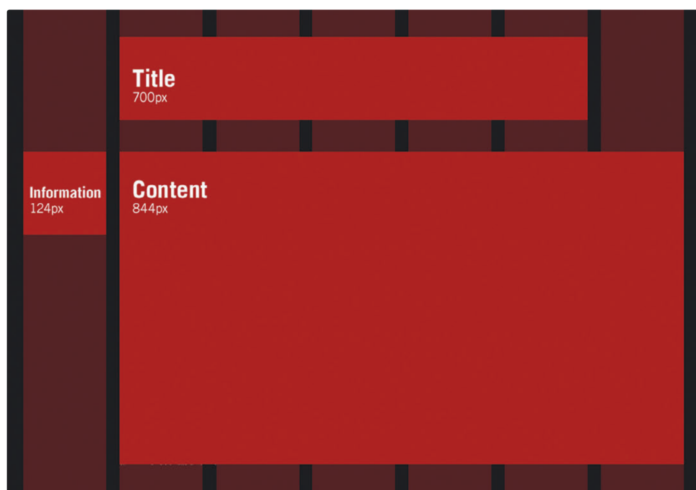
为了计算 <h1> 的相对单位，将 700 像素（目标）与 988 像素（内容）相除，得到 700÷988=0.7085。也就是 70.85%，我们可以修改样式表为。

```
h1 {
width: 70.85%; /* 700px / 988px = 0.7085 */
}
```

同时，对于外边距需要小心地处理，期望的左边距是 144 像素（目标），再次使用公式来计算相对宽度：144÷988=0.14575，相对宽度就是 14.575%，最后代码如下所示。

```
h1 {
margin-left: 14.575%; /* 144px / 988px = 0.14575 */
width: 70.85%; /* 700px / 988px = 0.7085 */
}
```

将这种方法应用到布局中的所有元素上，我们将会得到一个稳定、灵活、自适应的网格布局。这是一种非常简单的概念，但是在流式设计中带来高效的处理比例平衡的方法。有了这种技术，设计师就不能再抱怨维护比例是如何困难或者是怎样破坏他们布局的美感了。



一个简单的网格布局，只有一个标题区域、信息区域和一块很大的内容区。

设计师可以用不同的方式定义元素之间的边距值，其中一个就是计算边距的相对比例（在本例中，20 像素 ÷ 880 像素）。另外一个就是固定边距，在本例中，我们就设置为 20 像素。

每种方法都有其优点和缺点。使用相对的边距值，虽然能够达到较好的平衡比例，但是设计师就会承担在宽屏上边距过大的风险。固定边距虽然在边距上有一点不完美，却确保了在不同浏览器尺寸情况下一致的外观。

2.4　自适应的流式布局

尽管流式设计可以适应很多种不同的分辨率，但是仍有一个通病，在分辨率太低（比如手机或者 PDA 设备）或者太高的情况下，界面看起来就有点可笑。例如，一个三栏布局的页面在只有 240 像素的 PDA 屏幕中看起来就会非常糟糕。另外，如果设置了“最小宽度”，布局中可能会包含一个水平的滚动条，这让阅读变得很不流畅。

为了解决这个问题，可以使用一项技术来改善不同屏幕分辨率情况下的内容自适应问题，这也就是“适应性流式布局”名称的由来。接下来，我们将创建在 640 到 800 像素，320 到 640 像素，240 到 320，以及 240 像素以下分

辨率情况下只是稍有不同的定制布局。同样，定制的调整也能够分别用在 800 到 1024 像素，1024 到 1280，甚至 1280 像素以上的情况。

这样做的好处是显而易见的，设计师在能够切换不同布局分辨率的同时又保持了布局的准确。无论是最低分辨率还是最高分辨率，都能够很好地处理，同时保留了空间平衡的设计准则，而不用担心用户使用何种浏览器。

要建立这种类型的布局，需要准备两件事。为每种范围的分辨率准备分离的样式文件和一种获取用户屏幕分辨率的方法。第一件事情就是创建一些列可替换的布局文件。例如，一个文件命名为 narrow.css，在非常低的分辨率情况下使用。另外一个命名为 normal.css，在普通的分辨率情况下使用。第三个命名为 wide.css，专门应付那些高分辨率的情况。

这样做的好处是显而易见的，设计师在能够切换不同布局分辨率的同时又保持了布局的准确。

有了这些样式文件，我们就可以使用 JavaScript 来做一些简单的替换。例如，使用 Kevin Hale 的“动态布局技术（Dynamic Resolution-Dependent Layout Technique）”(particletree.com) 或者 Marac Van Den Dobblesteen 的“Switchy McLayout”（alistapart.com）。所有样式文件的声明和 JavaScript 脚本放在所有页面文件的头部。

```
<!-- Narrow style sheet -->

<link rel="alternate stylesheet" type="text/css" href="css/

narrow.css" title="narrow" />

<!-- Default style sheet -->
<link rel="stylesheet" type="text/css" href="css/normal.css"

title="default" />

<!-- Wide style sheet -->
<link rel="alternate stylesheet" type="text/css" href="css/

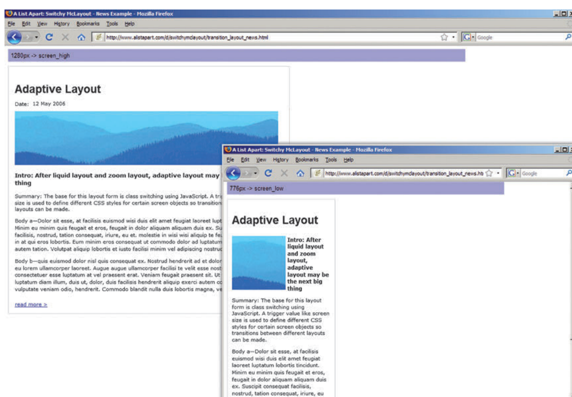
wide.css" title="wide" />
```

```
<!-- Included JavaScript to switch style sheets -->
<script src="scripts/dynamiclayout.js" type="text/
javascript"></script>
```

注意，在“narrow”、“default”和“wide”这 3 个外部样式文件链接中的属性。认真看一下源码中 DynamicLayout（）函数就能发现，可以很容易地根据样式表链接的属性进行样式文件的切换。下面来看看具体的例子。

```
   function dynamicLayout(){
    var browserWidth = getBrowserWidth();
    // Narrow CSS rules
    if (browserWidth < 640)
    changeLayout("narrow")
    }
    // Normal (default) CSS rules
    if ((browserWidth >= 640) && (browserWidth <= 960))}
    changeLayout("default");
    }
    // Wide CSS rules
    if (browserWidth > 960){
    changeLayout("wide");
    }
}
```

注意，因为适应性的流式布局依赖 JavaScript，所以需要客户端支持 JavaScript，这样脚本才能探测用户浏览器的分辨率，并且进行相应的切换。



图中是 Switchy McLayout 布局在不同浏览器尺寸下的示例。根据浏览器的尺寸，布局进行了自动的调整。

“支持变量的固定宽度布局”（Variable fixed-width layout）是由 Richard Rutter 开发的，基于 Simon Collison 布局的相似技术（在“Redesign Notes 1：Width-Based Layout”中进行过讨论，colly.com）。屏幕尺寸发生变化时，布局和字体也相应地变化。布局尺寸实时地发生变化，所以如果你调整了浏览器的尺寸，布局会随之变化以适应新尺寸。



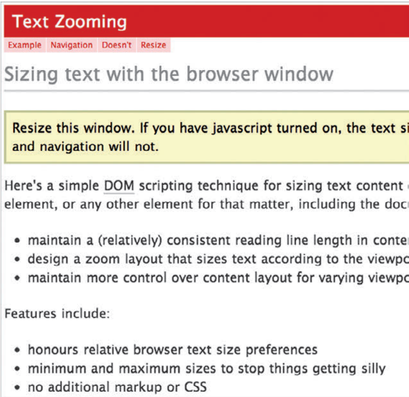
Richard Rutter 的基于变量的固定宽度布局使用 JavaScript 来自动缩放布局和字体。在 1280 像素分辨率下，是四栏布局；但是在 850 像素下，布局自动调整为三栏，第四栏移动到了所有栏目的下方。

就像之前提到的那样，流式布局的一个普遍问题是文本要么变得很长、要么

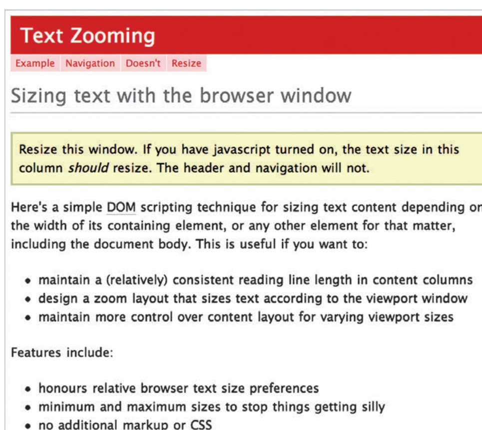
被挤到一团导致布局的可读性很差。在窄屏幕中，文本之间过大的空白是最大的问题，当然相反的极端情况也同样让用户很头疼。可以使用最大宽度和最小宽度来解决这个问题，但是这样的话，布局就被转换为宽度部分固定的布局，丧失了总体的灵活性。Tinned Fruits 的文本缩放技术（tinnedfruit.com）基于 JavaScript，根据用户浏览器的宽度自动调整文本的尺寸。当屏幕变宽时，字体就变大。相反，屏幕变窄时，字体就变小。另外，可以设置最大和最小的字体尺寸防止奇怪的情况发生。更进一步，设计师甚至可以决定哪些元素受到文本缩放的影响，哪些元素免受影响。

我们可以向网页中添加基于 JavaScript 的字体缩放。在页面中插入下面的代码，根据情况进行适当修改。

```
<script type="text/javascript">
    var contentZoom = new TextZoom(
    "Content", // Reference element
    "Content", // Target element
    0.22, // Zoom ratio
    70, // Minimum font size (%)
    130); // Maximum font size (%)
    addLoadEvent(textZoomInit);
</script>
```



字体缩放技术：左图是一个具有最大尺寸的页面，显示较大的字体。



字体缩放技术：和上图一样的页面，宽度调整为 700 像素。字体随着浏览器变化进行了调整。请注意导航菜单中的字体并没有发生变化。

在 Soh Tanaka 的文章“Smart Columns with CSS and JQuery”（sohtanaka.com）中，作者提出了一种类似的技术。为了获得最好的浏览体验，通过脚本来修改 DIV 的宽度，同时根据浏览器当前的尺寸决定显示列数的多少（根据浏览器尺寸的变化）。使用 JQuery 脚本清除所有列的多余空白，然后平均分配给现有的列。这项技术是自适应流式布局的一个好例子，特别适用于那些对浏览器不是一视同仁，而是自己调整浏览器尺寸的用户。

2.5 弹性布局

当浏览流式布局的网站时，内容区可能变得太宽而降低了可读性。为了提高可读性，设计师尝试通过 CSS 的属性来限制布局的最大宽度。弹性布局（或者称为可缩放）布局采用了完全不同的方法来实现相同的效果。这种布局的基本思想是根据用户的字体大小而不是浏览器视口大小来调整网页布局的宽度。

像素是计算机屏幕上无法缩放的一个点，em 则是字体的一个方形单位。根据用户对于字体尺寸的选择，em 单位经常发生变化（Elastic Design, alistapart.com）。相应的，设计师使用 em 单位，一种与弗雷元素尺寸相关的相对单位来表示字体的大小。通过为布局中的区块和字体使用 em 单位，设计师能够实现网页布局的整体缩放，保持内容区每行的宽度适中，实现接近现实的缩放效果。最后，当用户增加字体尺寸时，布局自动缩放，就像它们是一个有弹性的物体一样。

由于不依赖于浏览器视口的变化，弹性布局有点类似于固定宽度布局，同时具备了它的优点和缺点。设置很大的字体尺寸会扩展布局的宽度和高度，最后布局变得无法使用和阅读。但是，这种情况极少发生，因为很少有人会把字体放大 3 倍（特别是设计师已经使用了一个足够大的字体作为基准字体）。

像弹性布局一样，这种布局开始时最困难的一件事情就是计算合适的 em 值。为了简化从像素（px）到 em 的转化，可以设置 body 中 10 像素的字体基准尺寸为 0.625em 或者 62.5%（多数浏览器中字体的尺寸是 16 像素[1]，10 像素是 16 像素的 62.5%），这样在计算字体的子元素的时候更容易些。例如，设置页面主体部分字体尺寸为 0.625em，那么一个 960 像素宽的 DIV 就是 96em。

这些元素的存在增加了弹性布局的复杂性，因为在字体的尺寸发生变化时，需要维护布局中所有元素的比例关系。在一些情况下，弹性布局允许所有元素随着字体的增加而变化，也就是说，图片也会因为用户设置的不同，跟着布局的变化放大或者缩小。

当然，这可以通过为图片赋予类似 的百分比单位值来解决。但是记住，低分辨率的图片放大后会极大地降低质量。浏览器进行的是“模拟”放大，导致图片模糊并出现矩形块。另外，浏览器会从服务器加载整个（大）图片，并且进行缩放，增加了服务器的负载，提高了页面的加载时间。因此，多数设计师经常为图片使用绝对值。

关于弹性布局，或者基于字体尺寸设置宽度的布局，人们还非常关心如何展示多媒体内容，诸如图片和 Flash 对象。

在 Harvey Kane 的“Automatic Magazine Layout”（alistapart.com）一文中提到的另外一个解决方案需要用到一些数学和 PHP。文章标题取决于杂志中图片是如何被显示的：永远完美的布局和组织。看看下面的 PHP 代码。

```
# include the class file
require_once( ‘magazinelayout.class.php’ );
# Define the width for the output area (pixels)
```

[1] 16像素是一个多数浏览器默认的字体大小，但是很多人会将电脑中的dpi从120调整到96，这就会让系统中所有的字体看起来都比较大。

```
$width = 600;

# Define padding around each image; this *must* be included
# in your style sheet (pixels)

$padding = 3;
# Define your template for outputting images
# (Don' t forget to escape the &)

$template = '<img src="image.php?size=[size]&file=[image]"
alt="Screenshot" />';

# create a new instance of the class
$mag = new magazinelayout($width,$padding,$template);

# Add the images in any order
$mag->addImage( 'landscape1.jpg' );

$mag->addImage( 'portrait1.jpg' );
$mag->addImage( 'landscape2.jpg' );

# display the output
echo $mag->getHtml();
```

我们可以预先定义整个杂志设计布局需要使用的宽度。因此，如果能够决定用户浏览器的宽度，我们就能够计算出图片布局应该有多宽。这当然不难实现，因为在我们的第二个技术：自适应的流式布局中使用过。在这段代码中，Kevin Hale 使用了一个 getBrowserWidth（）的函数。感兴趣的读者可以看看他的文章，增加对这个方法的理解。

如果能够使用这个方法获取到用户浏览器的宽度数值，我们就能根据这个数值找到内容区域应该设置的像素值（或者用来存放图片的区域的宽度）。假设我们希望放置图片的内容区设置为 70% 宽。我们只需要通过简单的数学运算，来得到浏览器宽的 70% 是多少像素（px）。

像素宽度 = 内容区的百分比 × 浏览器宽度。

```
$width=0.70×getBrowserWidth();
```

当然，这个计算很简单，也是流式布局中处理图片的简单方法。在 PHP 脚本运行后，调整脚本来自动找到图片的像素宽度，这样就方便地为流式布局中的图片或者其他固定宽度的元素设置宽度。

弹性布局的最大优点就是具备始终保持设计元素比例关系的能力，以及确保可读性和定位准确。弹性布局是那些希望在流式布局和固定宽度布局之间妥协的设计师的首选。流式布局和固定宽度布局的优点在弹性布局中都有所体现。但是，这种布局实现起来非常困难，在完成之前需要大量的深入理解和测试。

现在，我们继续使用前面用过的 HTML 结构来实现一个简单的两栏布局。如果 1em 相当于 16 像素，那么 960 像素就是 60em。左边 600 像素是 37.5em，右侧 360 像素是 22.5em。为了方便使用，下面是计算公式：1em ÷ 16 像素 ×（像素宽度）=em 单位。

```
#container {
    width: 60em;
}
    width: 37.5em;
    float: left;
    display: inline; /* IE 5/6 Doubled Float-Margin Bug */
}
#rightCol {
    width: 22.5em;
    float: right;
    display: inline; /* IE 5/6 Doubled Float-Margin Bug */
}
#footer {
    width: 60em;
```

```
    clear: both;
}
```



不同字体尺寸下的弹性布局截图，随着字体的增大，布局的宽度也相应增加了。

设计社区中在热烈地讨论灵活布局的生命力。下一代浏览器，诸如 Firefox 3+、Opera 9.5 + 和 IE 7+ 都带来了一项似乎会减轻网站开发人员工作的特性——全页面缩放（Full page zoom）。

不仅是简单地增加网站中字体的尺寸，浏览器现在支持用户自然地缩放布局，包括布局中的图片和背景。整个页面设计都会根据选择进行等比例缩放。同时，所有的固定宽度布局也变成了“可变化的”。布局中始终包含内容的区域也支持这个特性。这样，就不会出现在上一代网络浏览器中经常出现的盒模型叠加的现象。直观地讲，这项技术将会导致弹性布局的荒废和消亡，因为他们达到了需要 CSS 代码才能实现的同样的效果。

然而，事实并非如此。正如 Zoe Mickley Gillenwater 所指出的（“为什么浏览器不能杀死弹性布局”，zomigi.com）。如果为用户提供一个带页面缩放的固定宽度布局，用户看到每行的字数会与设想的不一致（例如，一些用户因为视力的原因会选择大一点的字体）。

在固定宽度布局的页面中，可变的字体尺寸使内容的显示变得难以预测，同

时也很难保留设计中的比例和平衡。另外，当页面放大（或者缩小）时，某些情况下并不会放大 Logo 或者图标。换句话说，这种布局允许用户只缩放固定的区域或者元素，这在浏览器缩放中是无法实现的。当然，基于像素布局的所有缺点仍然存在，包括水平滚动条和过多的空白。最后，全页面缩放对满足不同用户需求有所帮助，但却不是开发人员的终极解决方案。

2.6　混合布局

现实中，设计师通常混合使用固定宽度、流式布局或者弹性布局元素，为用户提供每种方式的优点，并且将缺点降低到最小。

例如，内容区域通常会使用 em 单位（以此来确保每行的适当长度以及在 IE6 下的字体缩放）。侧边栏通常使用固定单位（因为侧边栏通常会包含固定宽度的 Banner，这样固定宽度才比较可行）。另外一个相关的技术是为流式内容指定“最小宽度”和“最大宽度”属性。使用 CSS 属性而不是弹性元素，让用户可以根据个人喜好调整每行的长度。

Jello-liquid 布局虽然不是很流行，但却比较有趣（“Jello：A Different Liquid Layout”，uwmike.com）。为了保持内容区的可读性，他努力降低布局增长时比例增长的速度。所以，一个自适应布局在 1024×768（960÷1024=0.9375）视口中具有 960 像素宽度，在窗口增长到 1440×900（1440÷1350=0.9375）时会得到一个比 1350 像素小的宽度。为了实现这种效果，布局中很大一部分元素都使用了固定的像素值，剩下的元素使用了与窗口宽度相关的相对值。

固定部分占布局的比例越小，布局随着窗口拉伸时所获得效果就越好。很简单、也很智能。

另外一个有趣的方法是结合了流式和弹性元素的流式弹性布局（fluid elastic layout）。这种布局基于 em 单位，但是使用相对值设置了“最小宽度”和“最大宽度”（考虑了窗口的宽度，赋予用户将布局限制在一定缩放范围内的能力）。通过设置最大和最小宽度，文本缩放到一定程度的时候会停止变化。布局随着窗口的变化而自动调整保留了适应性，同时也是弹性的。在窗口尺寸或者字体尺寸发生变化时，各列的宽度在自动变化。

在下面的例子中，为容器和每一列设置了“max-width”属性。这样布局就会限制在整个视口的宽度。同样，在较老版本的 IE 中并不支持“max-width”，所以我们需要为 IE 做一些特殊的处理，代码如下所示（在下一个例子中将不再列出）。

```
#container {
    width: 60em;
    max-width: 100%;
}
#header {
    background-color: #cccccc;
    width:60em;
    max-width:100%;
}
#leftCol {
    width:37.5em;
    float:left;
    display: inline; /* IE 5/6 Doubled Float-Margin Bug */
    max-width:62.5%;
}
#rightCol {
    width:22.5em;
    float:right;
    max-width:37.5%;
}
#footer {
    width:60em;
    max-width:100%;
clear:both;
}
```

2.7 哪种布局最适合您的网站?

固定宽度的方法通常在带有大量图片的高级视觉布局中表现良好(例如娱乐、促销或者基于 Flash 的网站，还有一些作品集页面)，这样图片能够通过精确的定位实现更加准确、较少错误的设计。实现这种布局，设计师需要考虑用户的平均情况，确保设计的布局对大部分用户不会显得太窄或者太宽，在不同浏览器间保持良好的可用性。编写固定布局同样比较规则和直接，开发人员不需要为各种元素之间的关系和平衡耗费太多心思。

对于哪种布局最合适不会得到标准答案。限于使用环境、时间限制以及设计师的技能，每种布局都有特定的应用情况。

如果这正是你的想法，也许你会考虑在设计中使用流式布局。这种布局制作非常耗时，也不是很容易开发，并且在一些不可预知的环境中存在一定风险，但是提供了用户根据喜好调整布局的特性。流式布局具有较高的生产成本，但是显著降低了维护成本，同时为灵活性和跨分辨率的网站提供坚实的基础。设计师只需要确认内容区具有合适的行宽，并且在必要的情况下使用“min-width”和“Max-width”。

是否仍然犹豫不决？弹性或者部分弹性的设计是另外一个选择。在正确使用的前提下，弹性布局在带来了更多可预测性的同时保持了布局的灵活性。因为弹性布局依赖于字体大小 (而不是浏览器的窗口尺寸)，弹性布局允许设计师“冻结”布局中区块间的比例，确保元素之间的平衡和良好的可读性。特别是在文本较多的网站中，保持良好的可读性的要求非常严格 (固定布局中也一样重要)。弹性布局更多的是用在杂志、在线商店、博客或者类似的网站上。聪明的流式网格使用者可以创建既忠实于原始设计比例和经典图片设计原则，又保持适应性的布局。

一些设计师希望在不同浏览器之间获得完美的一致，流式设计的支持者们却认为布局并不需要在不同平台和分辨率下保持一致。

实际工作中，设计师经常为字体和容器使用 em 单位，而其他元素则灵活地使用像素和相对值来建立一个弹性的布局。这些或者相似的混合布局在实际情况中最常见，反映了设计师希望为每种情况找到最佳解决方案的努力。

在总结之前，有必要提一下智能手机、上网本和游戏终端，这些设备现在都可以进行网页浏览，所以更低（或者更高）的分辨率变得越来越重要，而且应当进行认真考虑。对于这类平台，动态的分辨率切换技术将会非常有用，而且在将来扩展你的布局类型时会是不二的选择。虽然构建自适应性的布局需要耗费很多时间，但是由于能够处理或小或大的各种类型的分辨率，会帮助设计师为用户呈现最合适的效果。随着屏幕宽度的不断增加，使用所有这些技巧只是个时间问题。

排版：原则、方针和常见错误

第 3 章

Alessandro Cattaneo　Yues Peters　Jon Tan

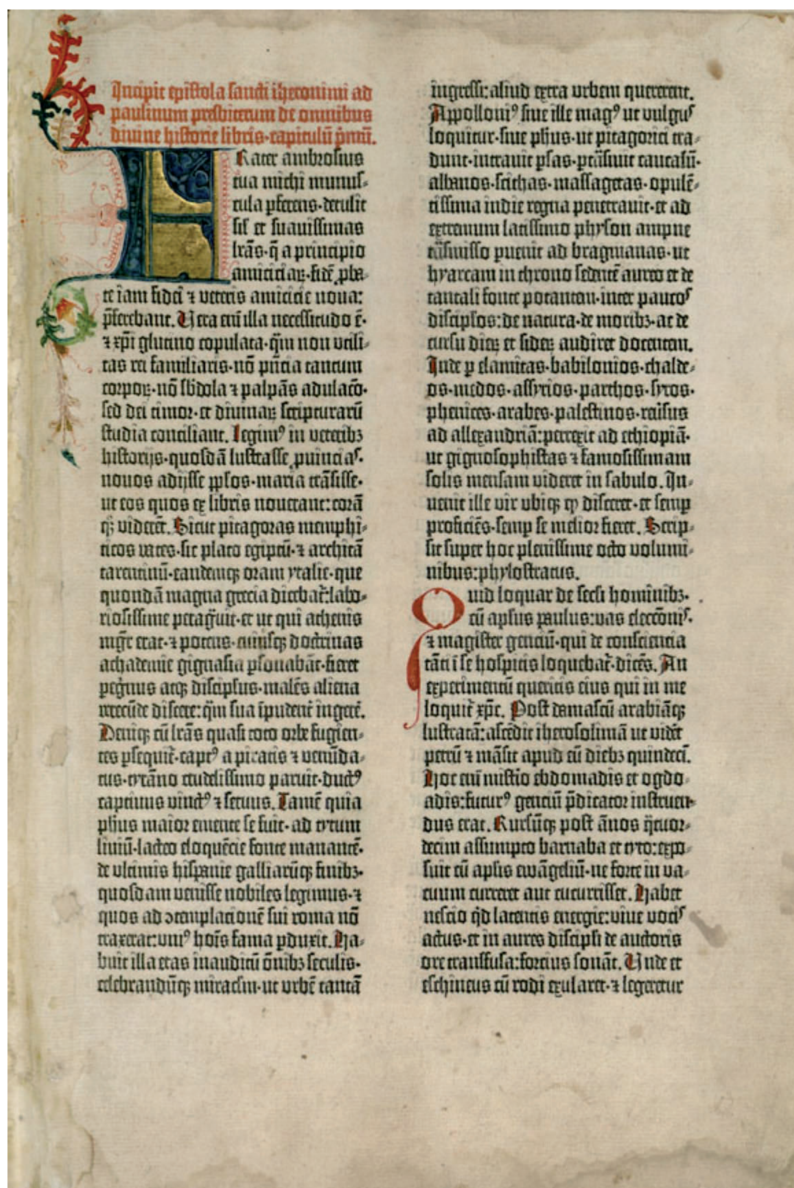
排版是设计的灵魂，是视觉识别能力的核心。这章将对网站排版进行探索，并描述组合的方法和技巧，它们将给予文档生命并使文档容易理解。

Robert Bringhurst 说“排版将使内容增色”，他是著名的排版经典之作——《排版基础》（The Elements of Typographic Style）的作者。网站排版尤其是文字选取、排列和字体设计，丰富了文本的意义，同时提供了一个使文本获得生命的框架。

排版包含了很多主题和应用，甚至包括现在互联网上的信息数字化。它包含选择最好的字体、提高可读性、定义品牌和企业形象、销售产品、使信息更容易被读者理解。如果排版的一些基本原则应用得不到位，有时足以导致读者流失。

让我们把目光投向排版刚刚兴起的时候，古登堡（Gutenberg）在 1455 年使用的黑色字体，也许在今天并不合每个人的口味，但是毫无疑问即使按今天的标准，它也堪称做工精美。当为网站进行设计时，我们的线条无法如此精美，我们的连字（ligature）不能像 400 多年前古登堡在纸和羊皮纸上印的那样。

设计产品的方法已经发生了改变，但是并没有完全改变。文字排印不再是由专业技师完成的手工劳动，更多是根据设计师设定的视觉模型通过命令进行抽象。字体不再是显示在纸上，而是被光栅化成像素，显示在电脑屏幕上，这是网站排版不同的地方。不再是挑选印刷的纸张和材质，我们现在要处理的是不同的显示器类型、分辨率、浏览器和操作系统（很多系统有不同的引擎）。网页排版时必须把它们都考虑在内。



Incipit epistola sancti iheronimi ad paulinum presbiterum de omnibus divine historie libris. capitulum primum.

古登堡圣经第一卷的第一页，由 1455 年左右的早期字体印制。装饰性的彩色首字母是由抄写员手绘的。

现在只要鼠标轻轻点击就可以使用大量的字体。这并不是说它们都适合用在网站上。大部分字体是供印刷使用的，而不是供我们用来浏览网页的、分辨率相对较低的屏幕使用的。专门为屏幕设计的字体问世和对跨平台和跨浏览器支持的要求，把我们放在了一个和昔日设计师和印刷业者相似的位置。 我们仅仅可以使用一小部分字体，我们对它们已经非常熟悉，有时在一些 CSS 样式的帮助下，我们会尽可能广泛地使用它们。

网站设计和桌面出版都在进化演变，但道路不同——网站设计还处在萌芽阶段。一些东西已经消失，而一些东西我们已经拥有。我们可以利用现有的工具很好地工作，比如 HTML 和 CSS（如果非常有必要，JavaScript 和 Flash 也可算在内）。一些排版中的旧规则和指导原则可以应用于网站，但不是所有的都可以。在线交流是一种不同的新媒体，通过它，编辑和读者可以交流，这将会影响到排版。

3.1　传统排版和网站排版

我们经常认为写作是一件简单的事情；我们经常认为使用正确的语法、决定元素在页面布局中的顺序以及选择视觉效果好的字体就足够了。但是排版的艺术包含很多经常被忽略的因素，对于网站而言更是如此，整个写作和发表过程是沉浸式的，文本的质量经常被忽略。在这一章，我们将会展示很多不同的排版技术、良好的实践和一些经验之谈，同时也包括常见的排版错误。但是首先让我们仔细看一看网站排版的一些基本概念。

清晰度受字体设计、屏幕优化和微观版式影响。它与彼此可以相互识别单个字符、字母或字形相关。我们不能改变字体的外形，但是可以选择正确的字体，并且选择相应的版式。当涉及清晰度问题时，许多设计师在一些标准、探索法和排版技巧上达成了共识。比如，大写字母会比小写字母更难以扫读，正体字比斜体字更容易识别。在正文文本和背景间合适的对比会提高清晰度，正如适宜的行宽和行高所产生的效果。与字母的下半部分相比，上半部分对于扫读的意义更大，然而升部（ascenders）和降部（descenders）的高度对于字母的识别具有重要作用，这也是很多人认为有衬线（serifs）字体更容易识别的原因。正如你所看到的，当我们谈到清晰度时，很多因素都在起作用。

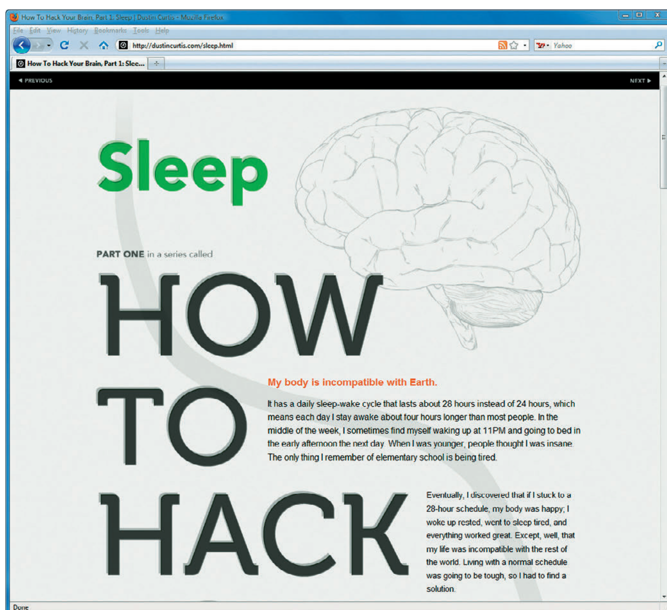
A SPECIMEN

By WILLIAM CASLON, Letter-Founder, in Chiswell-Street, LONDON.

This SPECIMEN to be placed in the Middle of the Sheet ; U u, Vol. II.

18 世纪著名英国字体设计师 William Caslon 设计的罗马字体的一个样本。

可读性是设计主要关注的问题。它受布局、内容块的摆放和独立栏目的展现影响。我们可以通过很多方式保证可读性，其中很多方式都受读者习惯影响。从根本上讲，人们“读”网站内容的方式和读印刷品的方式不同。理解这种行为和为这种行为进行设计非常关键，这也是用户体验设计的要点。网站排版需要深思熟虑。



对于一些设计者而言，网站设计完全就是排版。在 Dustin Curtis 的网站上，版式非常醒目，它蕴含着微妙的视觉线索。

行宽（measure 或 line length）就是每行文本的长度。最佳的长度取决于上下文，但是对于左对齐的文本块，45 到 75 个字符（包括标点符号和空白）之间的长度值得推荐。66 个字符被认为是最佳的长度：这个长度阅读起来最舒服，使眼睛不必在本行结束和下一行开始之时移动太远的距离。不要让行宽太短，因为频繁地在行与行之间移动，同样会使眼球疲劳。

令人惊讶的是，我们最近关于版式设计[1] 的研究发现，包含大量文本、排版相关的网站（新闻、杂志和排版相关的博客等）有更长的行宽，通常在 70 到 100 个字符之间（每行最多的字符数）。总体上，每行的字符在 75 到 85 个之间。不过，目前还没有研究表明这样的行宽是否会带来更好或更坏的可读性。

另外，很多网站是多栏布局。行宽在多栏布局中可能会短一些。根据习惯，对齐的文本通常在区域的下半部分；但是不需要精确控制单词间的空白和

[1] http://www.smashingmagazine.com/?p=8182

连字符，较长的行在浏览中的显示效果会更好一些。



Trentwalton.com，一个由 Trent Walton 设计的关注排版的博客。

虽然设置正确的行宽很重要，但是可读性可以通过正确的行高（leading 或 line height）显著提高。行高就是一行文本的垂直高度。它是两个连续文本之间的距离，或者说是两行文字基线之间的距离。为了防止一行的降部覆盖下一行的升部，必须保证足够的行高。在默认情况下，浏览器会把行高设定为 1em，或者字体大小的 1 倍，这个高度通常会显得太小以至于不好辨析字体。一个被广泛接受的经验是，把段落的行高设定为字体大小的 1.5 倍。实际上，根据我们对平均行高的研究发现，这个约定俗成的行高仍然在被广泛使用（平均行高是字体大小的 1.48 倍）。作为一个简单的指导，行宽越大，行高也应越大。

注意到我们在 CSS 中可以通过无确定单位的整数来定义行高值。这个数字作为度量因素和字体大小相关，并确保在文字大小改变后，行高也发生相应的变化。总体来说，无衬线（sans serif）字体比衬线（serif）字体需要更大的行高，页眉比正文需要更少的行宽。同样，较粗字体比较细的字体需要更大的行高。

字母间距（tracking 或 letter spaceing）是字母之间的间距，也是文本块之间的密度，它可以根据行宽进行调整。在分辨率较低的情况下，根据像素值调整字符间距可能有困难，因为它产生了不连续的效果。 正因为这样，字符间距经常用来突出副标题。然而，作为一个简单的指导，如果行宽越长，你

就越需要调整单词间距，字母间距也通常需要增加。要小心操作。一些衬线字体的字符有时可能会重叠，它们应该被当做连字处理。

当我们对字母间距进行实验时，要确保“AV”、“ft”、“co”清晰可读，并检查是否能区分出“vv”和“w”：即使是最小程度的不精确，也可能会使文字完全不可读。

成对字符间的字母负间距也被称为 kerning，它对于调整“ff”、“fl”、“ffl”这类字符组很有必要，它们实际上应该被当做连字。在很多情况下，负字符间距对于提高字体的外观很有用。比如，把“T”和“e”靠得更紧凑些。

单词间距（Word Space）可以根据行宽和行高进行调整。行越短，单词间距越小。较长的行通常会通过更多的单词间距受益，尤其是它们有相当可观的行高时。

排版色彩不是染色或上色，页面看起来应该连贯。如果你在屏幕上正视或眯眼看一块文本区域时，它应该有一个统一的色调，而不应该令人眼花缭乱。如果这样，阅读将会不连贯。只有在需要故意引起注意的情况下才会这样做。颜色也会受负字母间距影响。不幸的是，现在因为浏览器无法正常支持它，还无法对此进行调节。

3.2 基本的排版术语

这里是一个与排版相关的术语的简要总结，[1]以便我们对这篇文章接下来的部分有一个一般性的认识。

重音（Accent）：一个在字母附近或横穿字母的附加符，表示了发音的变化。比如 ç, à, ò, é, Å。

中轴（Axis）：一条真实的或想象的直线，字母围绕它旋转。

基线（Baseline）：一条想象的线，字母停留在它上面。

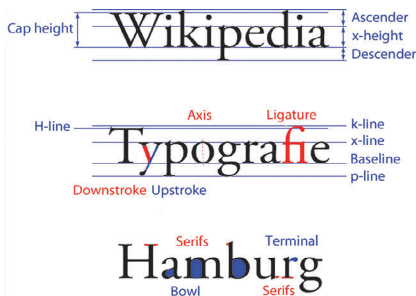
凸型（Bowl）：一些字母的圆形或椭圆形部分，它构成这些字母的主体，比如，大写字母 C、G、O，小写字母 b、c、e、o、p。它相当于眼眶。

[1] 这个总结部分基于TypeNow的术语表 (http://www.typenow.net/glossary.htm)。

大写字母高度（Cap height）：那些大写字母的高度。

围白（Counter）：被字母形体包围的空白，无论是完全被包围（比如在“d”或“o”中）还是部分被包围（比如在“c”或“m”中）。

横笔（Crossbar）：在大写字母中连接两个笔画的一个水平笔画，参考字母“A”和“H”。



一个来自维基百科（Wikipedia.de）的排版术语的概要。

段首字母下沉（Drop Cap）：段首字母大写，并占据多行的高度。

左对齐（Flush left）：使任何空白都出现在右边，文字都靠左对其。也被称为 ragged right。

右对齐（Flush right）：使任何空白都出现在左边，文字都靠右对齐。

字形（Glyph）：字体中每一个字符（比如：G、$、?、7）都会由一个字形代表。一种字体中，一个字符可能会有多个字形。它们通常被称为替代字符。

连字（Ligature）：两个或更多的字母连接在一起，通过空间交互组成一个独立的字母。

大写（Majuscule）：大写的字母。

无衬线（Sans serif）：没有衬线的字体。

衬线（Serif）：在字形每一个主要的笔画后面的一个装饰性小笔画。有衬线的字体被称为 serif 字体，没有衬线的字体被称为 sans serid 字体。

小型大写字母（Small Caps）：小型大写字母就是高度较矮的一类大写字母。很多应用程序通过压扁大写字母产生小型大写字母，但是这样产生的字母缺乏合适的重量和比例。一个真正的小型大写字母保持了符合比例的重量，但是却以更小的尺寸显示字母。

终笔（Terminal）：无衬线字体的最后一笔。

x 高度（x-height）：小写字母的高度，通常以小写字母 x 为例。

实际上，在指字体时人们经常把“typeface”（字体）和“font”（字体文件）混用。typeface 指的是某类特殊字体的设计，而 font 指的是包含我们使用字形的那个文件。比如，George 指的是 typeface，而 Georgia-Bold.ttf 指的是 font，一个包含了这个字体所有粗体字形的文件。换句话说，对字母、数字和符号等的实体化集合（无论是金属块还是计算机文件）就是一种“font”（你选择并使用的文件）。

当我们指字体集合的设计时（也就是字体看上去的样子），我们指的是“typeface”。打个比方，我们播放 MP3 文件，但是我们听到的却是优美的歌曲。

3.3　相对和绝对的字体尺寸单位

也许数字化排版最不同的特点就是用户能够根据自己的偏好调整字体大小。当选择了一个字体大小后，网站开发者可以使用绝对或是相对的度量单位，也可以在 CSS 中设置。总体来说，绝对单位更适合印刷品的设计，把它们用于网站排版不是很好的实践。唯一的例外是打印样式表，绝对单位可以帮助定义文档上设计区域的大小。目前被支持的度量单位包括 pt、pc、cm、mm 和 in。但是像 xx-small、x-small、small、medium、large、x-large 和 xx-large，也同样可以选择。

设定“绝对”的字体大小并不意味着用户不能增加或减少字体的大小。现代浏览器允许大量针对文本的操作。甚至，绝对单位使设计者在父元素和子元素之间设定比例变成了不可能的事情，即使用户手动地增加或减少字体的大小，字体的大小也会不变。因此，“absolute”属性与样式表中元素的定义相关，而与屏幕显示无关。

设定“绝对”的字体大小并不意味着用户不能增加或减少字体的大小。

现代浏览器会默认把 <body> 元素的字体大小设定为 16 像素。如果你不在样式表中明确地指定其他值，将会使用这个值。为了避免复杂的计算，Richard Rutter 建议把 <body> 的字体大小设为 62.5%，这将把文字的大小降低到 10 像素（16 像素 ×0.625）。相应地，当把标题字体大小设定为 18 像素时，使用 1.8em，而不使用 1.125em（18 像素 : 16 像素 = 1.125）更合适一些。

实际上，字体大小通常使用相对单位，比如 px、em 或 %。px（像素）与显示器的分辨率相关。分辨率越高，像素密度越大。通常这也意味着字体看上去会更小。

“em”是通过你使用的字体大小来定义的度量单位。[1]它代表了基线（baseline）之间的距离，当字体属性被设定为 solid（不含行高）时。em 是一个相对的单位，它的值一般是文本元素父元素大小的倍数。所以，1em 对于 16 点（point [2]）的字体而言，意味着 16 点。如果用户浏览器默认的字体大小是 16 像素，那么 1em 就代表 16 像素，这基本仅仅是文本显示尺寸的一个属性值。如果用户把字体尺寸降低到 14 像素，那么 2em 将代表 28 像素。

使用 em 作为字体大小单位的好处是，这个度量单位在设定父元素和子元素的比例方面有着天生的能力，以及对 CSS 继承特性的利用。利用这个特性子元素可以继承父元素的大小，除非明确声明了进行尺寸更改。 比如，如果默认字体大小是 16 像素，<body> 元素的字体大小为 2em，<body> 之内的页眉字体大小为 1.5em，那么页眉的字体大小将是 <body> 元素的 1.5 倍，也就是 1.5×2×16 像素 =48 像素。

在某种程度上，百分比近似于 em 值：CSS 中文档的根、继承和父子元素的相对关系会为每一个元素定义大小。所以在上面的例子中，可以定义 <body> 元素的字体大小为 200%，页眉的字体大小为 150%, 结果是相同的（48px）。

在一些环境中，将最小宽度和最大宽度属性设置为 em 或百分比对于获得一个可读性好的行宽会很有帮助。通常，字符的平均宽度是 2/3 em。因此 30em 和 50em 可以被视为理想的行宽。

因为 IE6 不允许用户按像素缩放文本，因此建议使用 em 或百分比值。不过，当使用相对单位时要小心谨慎。使用一些尺度和分级（参考下文）来确保不同重量的文本元素能够准确、恰当地呈现。

因为 IE6 不允许用户按像素缩放文本，建议使用 em 或百分比值。

---

[1] 最初，em被认为等于某个特定字体中大写字母“M”的宽度，因为“M”在印刷过程中通常占据整个方块区域。然而在现代字体中，字母M通常比1em的宽度要略窄一些。另外，因为一些字符集不包含大写字母“M”，比如中文字符和阿拉伯字，今天em一般指相关字体的高度。

[2] 译者注：point是印刷中常用的度量单位，在英国和美国1 point = 0.351mm，在欧洲1 point = 0.376mm。

其他相对度量单位在实践中很少被使用：虽然被广泛支持，但“ex”和“smaller”、“larger”（它们把字体大小降低和增加 1.2em）关键字从来不会在样式表中被大量使用。同样，点（point）在印刷中很有用，但是也不会被用在屏幕上。人们浏览器的默认设置也许会略微不同，这可能会对版式产生负作用。为了确保独立于浏览器展示文本，使用 CSS 全局重置，允许你在一个安全的独立于浏览器的环境中，设置字体和设计网站的全部元素。有很多种 CSS 重置样式表，从 * { padding: 0; margin: 0: } 到比较长的那些，比如 Eric Meyer 的 Reset Stylesheet 4[1]，各不相同。

3.4　空白是最好的空间

在网页排版中使用空白的重要性是无法低估的，空白使正文的连续文本能够呼吸，并帮助读者获取网站传递的信息。空白也被称为“负空间”，是构图中元素之间的空隙，或者设计中没有被使用的部分，包括图形、外边距、页面和分栏（宏观空白）之间的空白，以及文本行之间、单词之间和图片标题之间（微观空白）的空白。当然，空白并不一定是白色的——这个词来自于印刷领域，因为在这个领域中，白纸即代表了空白。

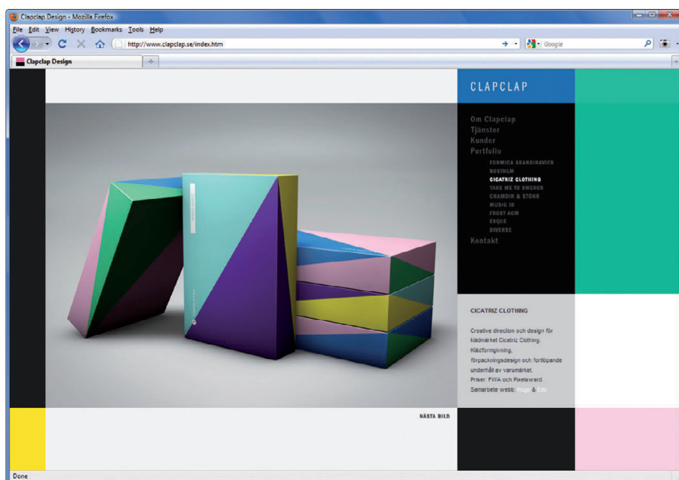


对排版细节的重视，尤其是空白，在 Elliot Jay Stocks 的展示网站上尤为突出。注意发挥空白的平衡作用，它用来突出引用内容和段落中的非活动空白。

1　http://meyerweb.com/eric/thoughts/2007/05/01/reset-reloaded/。

那些把用户的注意力从一个元素引导到另一个元素，组织布局并帮助建立信息结构的空白被称为“活动空白”（active white space）；那些把文本规划成块的空白被称为“非活动空白”（passive space）。

实际上，很多清晰度和可读性问题都可以被解决或至少被减弱，只要审慎地、均衡地使用空白。正像页间、内边距、外边距、行宽或行高的一些额外的像素可以很大程度上影响内容整体的清晰度。不仅如此，引文、图像和列表的缩进不仅可以改善布局的结构和展现效果，也可以使设计更富有动态和表现力。



在 ClapCalp 的展示网站，空白起了主导作用。它突出了所展示的产品的质量，使它们更加优秀、更加人性。

如果在内容和环绕内容的空白之间没有好的平衡，文本将很难阅读或扫读，使用户有种挫败感，并对你的信息缺乏信心。对于网站尤其是这样，很多设计者通过“above the fold”[1] 来填充空隙，试图抓住用户的注意力，这导致网站内容过于密集、难以使用，这当然不是设计者的最终目的。从用户的角度，空白提供了线索和定位，有助于产生自然的、令人满意的阅读体验。

一些设计者声称，设计永远不要使用太多的空白。事实并不总是这样，因为文本块的聚合和连贯对于一个平衡的布局是很重要的。平衡正空间（非白色

[1] 译者注：above the fold是一个平面设计术语，通常指报纸版面上第一版折叠线以上的重要阅读部分。在Web领域，一般是指首页第一屏你所能看到的、不需要向下滚动的那部分内容区域。

的）和负空间对于产生美丽、和谐和令人愉悦的版式至关重要。正如 Mark Boulton 在他的一篇文章[1]中陈述的那样，“一旦你知道如何设计和操控外部、内部和内容周围的空白，你将能够给读者一个好的开始、使他们更加精确地定位你的产品，甚至也许会用新的眼光看待你的内容”。

3.5　排版和网格

排版网格是由一系列垂直和水平的轴线构成的二维结构，它用来使内容结构化。当使用得体时，网格可以作为设计师以一种合理的、自觉的、自然的方式组织文本和图片的支架。它是看不见的灵魂，它会产生韵律、秩序和连贯，它经常被设计师用来更好地预见信息将被放置在哪里，以及使创意变得理性化。当图像元素必须快速而有序地合并时，也可以使用网格。平面设计中，网格表现出了最具生命力的意志。



Blueprint 是许多 CSS 框架中的一种，它使设计师能够更容易地使用网格开发网站。与此类似的还有 Typogridphy（基于 960.gs）、YUI Grids 和 YAML。

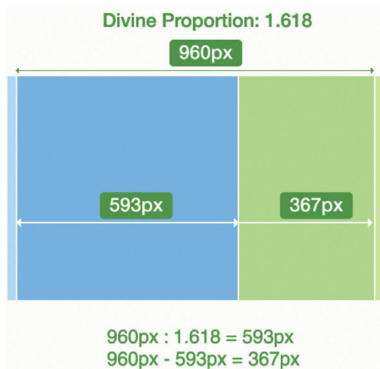
不设计元素该被放置在哪里是靠直觉来决定，网格在一个固定的二维结构中，允许元素精确地定位，回答了设计元素或文本该被放在哪里的问题。网格，

[1] http://www.alistapart.com/articles/whitespace/。

很多设计师认为它是内容之前的先决条件。挑战是在内容将被构建在其中的网格和内容的主要特征之间找到合适的平衡。一些人说网格限制了创造力，然而另一些人把它当做一个强大的框架，它可以让设计者做出完美、精确和准确的设计。

当基于网格进行设计时，设计者通常从空白画布或白纸开始。为了寻找合适的网格，他们使用了一般的布局规则和公式（比如黄金分割、三分法等）来把白纸分成适宜的部分，并且选择内在的、令人满意的页面和分栏比例。最后，通过对外边距进行试验来创建一个完美的、动态的排版结构。剩下的是用内容来填充结构，用内嵌在网格中的严格规则来限制设计。

有意思的是，正如 Mark Boulton 在他的文章“通过简单 5 步来设计网格系统”[1] 中陈述的那样，“作为结果，这样的尝试意味着那些被设计出来的美丽事物在本质上更可用”（也被称为美学可用性效果）。一个平衡的构图自然在美学上更令人喜欢、更好用，更能引导你和用户进行沟通。



使用了黄金分割的布局构图。

黄金分割可能是最常用的网格布局方式。为了给访客提供一个令人满意和直觉性的布局，考虑一下纸张的分割方式。黄金分割是等于 1.618033988749895（≈ 1.618），是彼此相邻放置的事物的长度比值。它也被称为 Fibonacci 序列。这个比例的强大之处在于，你在宇宙的各个角落都可以发现它。

考虑上面的例子。你可能想增加布局的固定宽度。布局宽度是 960 像素。你想要大块的内容(<content>)和小块的边栏(<aside>)，该如何计算栏目的宽度?

- 首先计算 <content> 版块的宽度。确保版块和整个图层的宽度比例为 1.62。因此，可以把 960px 除以 1.62，将近似得到 593 像素。
- 从图层整体的宽度中减去 593 像素，你将得到 367 像素。
- 现在，如果计算 <content> 版块、<aside> 版块的比例（593 像素：367 像

[1] http://www.markboulton.co.uk/journal/comments/simple_steps_to_designing_grids/。

素 ≈ 1.615）与容器宽度、内容版块宽度的比例（960 像素 : 593 像素 ≈ 1.618），你会发现其比例差不多都是相同。

这就是黄金分割背后的思想。对流动和弹性布局同样适用。

当然，你不必遵循黄金分割。连贯协调的结构，这个结果是最重要的。黄金分割只是实现它的一种方式。另一种选择是流动网格布局，它在讲布局的那一章已经讨论过。

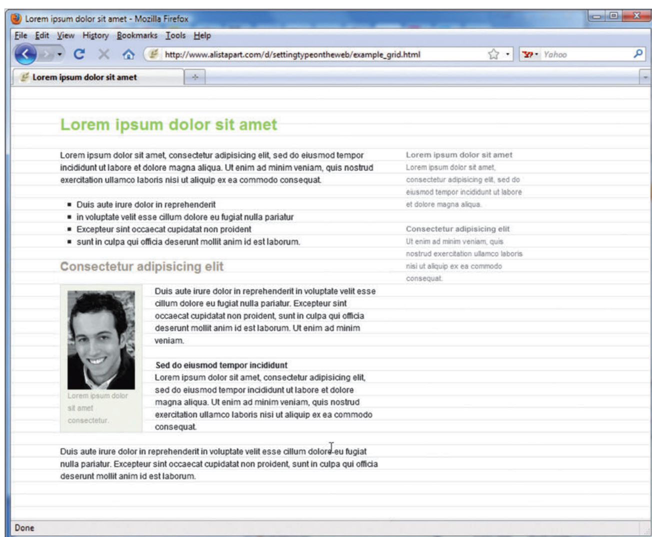




使用或者不使用网格：一个美丽的基于网格的设计，在排版上煞费苦心。由 Mark Boulton 和 Khoi Vinh 设计，为了展示“网格，妙不可言”（http://www.subtraction.com/2007/03/18/oh-yeeaahh）。

3.6 垂直节奏

作为排版人员，Robert Bringhurst 说到，“排版中的空隙，如同音乐中的节拍”。它是无限可分的，但是一些比例合适的间隔比任意数量的无限选择更有用。在音乐中，对节拍的规则使用构成了韵律；在排版和网站设计中，规则地使用空隙引出了网格理论和垂直节奏理论。正如音乐中的和谐可以通过一定比例的时间间隔获得，排版中的和谐可以通过一定比例的空间间隔来获取。



Wilson Miner 的“通过基线网格进行网站的排版”[1]中提到的垂直节奏。

为了理解垂直节奏，想象一下经典的布满线条的练习本。每条线（也被称为基线）彼此等距。如果文字正好位于基线上，意味着每一行文本的基线正好和每一行的基线吻合，这时可以说设计获得了连贯的垂直平衡。这不仅适用于主体内容，也适用于标题、脚注和边注。现在，如果我们把画好线条的纸放在布局的下面，把它当做一个看不见的排列文字的样式指导，你就获得了布局的垂直节奏。很显然，按垂直节奏排列的文本更容易扫读和阅读，因为它们似乎“呆在”正确的位置上。

[1] http://www.alistapart.com/articles/settingtypeontheweb/。

维护垂直节奏的关键是行高，换句话说，是两行基线之间的距离。在 CSS 中，行高（line-height）属性基本上设定了整个文本流中的网格，给用户发出了一个看不见的提示，以便他们能够顺畅地阅读。因为垂直节奏和字体大小有关，使用相对单位 em 值作为行高是很自然的。一个好的副作用是维持了设计中页面布局的平衡，而不用管所用字体的大小、样式和种类。

为了维持 CSS 中垂直节奏，元素间的空隙和线条间的空隙需要使用尺寸相等的基线网格。例如，如果使用 18 像素的基线网格，意味着每对基线之间将有 18 像素，这意味着线条之间的间距也应相应地被设置成 18 像素。并且，每段的距离也应被设置成 18 像素，如下例所示。

```
body {

font-family: Arial, sans-serif;

font-size: 0.625em;

/* The font-size is now 0.625 * 16px = 10px. */

line-height: 1.8em;

/* The leading is now 18 pixels. */

}
```

如果希望在段落之间留一条空白直线，你需要定义一个合适的外边距，当然它要和基线一样高。否则，段落将不会“呆在”正确的位置上。

```
p {margin-bottom: 1.5em; }
```

当然，你可以通过调整垂直空隙以提高页面动态效果，并使文本流出其不意。然而，每一个变体都应该用头脑中的垂直节奏定义，并以行高为基础。同样，其他文本元素，如标题、脚注、图像（这也是文本元素？）和图像标题等应该总是基本行高的倍数。这种关系可以通过以下公式表述。

基线高度（basic line-height）÷ 字体大小（heading font size）= 行高（heading line height）

注意，在开发过程中，创建包含网格线的背景图片会起到很大的作用。在实践中，垂直节奏并不总是可以实现的，尤其是在处理很多不同尺寸图

像的时候。然而，它仍然是一个值得追求的目标，对改进页面效果非常有帮助。

Typographic scale

ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab
6 7 8 9 10 11 12 14 16 18 21 24 36 48 60 72 84 96 110

传统的排版分度。

3.7 分级（Hierarchy）和分度（scale）

每一个版式都需要分级。它的核心是，分级支持通过提供关于文本结构和组织的视觉线索而获得清晰度。分级定义了内容如何被阅读。它为用户展示了到哪里开始阅读和到哪里停止阅读。它把页眉和正文区分开。虽然通过文本块的颜色可以在页眉和正文间产生对比，分级则通过元素的大小显示出它们的不同。分级在使布局易于扫视方面起到了重要作用，它也是使网站排版变得可读的一项重要技术。正如 Jeff Croft 在他的“优雅的网页排版”中陈述的那样，传统意义的上的排版由分度组成。一个版式中字体大小的不同是很有意义的，理解这一点很有帮助。为了在排版中获得良好的分级，需要使用分度，下面是传统的分度。

6、7、8、9、10、11、12、14、16、18、21、24、36、48、60、72

你也可以小心翼翼地创建自己的分度。分度使文字和谐与连贯。另一种选择是，一些设计者倾向于在定义文本和标题的字体大小时使用 Fibonacci 数列[1](比如：16 – 24 – 40 – 64 – 104)。

下面是一个简单的关于排版尺度的例子，在 CSS 中定义如下所示。

```
body {

font-size: 0.625em;
```

---

[1] Fibonacci数列是一个特殊的数列，这个数列从第三项开始，每一项都等于前两项之和，比如：16 + 24 = 40; 24 + 40 = 64。随着数列项数的增加，前一项与后一项之比越逼近黄金分割0.6180339887。因为它们有助于设计者达到黄金分割，它们经常被用在更强调审美功能的设计和布局中。

/* 因为浏览器的默认字体大小是 16 像素，我们现在要把页面文字大小降到 10 像素，这样更容易计算 */

}



AIGA 网站通过对颜色、字体大小和行高的巧妙使用，达到了美丽、清洁和自然的分级效果。

```
p { font-size: 1.4em; /* 14px */ }
h1 { font-size: 4.8em; /* 48px */ }
h2 { font-size: 3.6em; /* 36px */ }
h3 { font-size: 2.4em; /* 24px */ }
h4 { font-size: 2.1em; /* 21px */ }
h5 { font-size: 1.8em; /* 18px */ }
h6 { font-size: 1.6em; /* 16px */ }
```

当然，要在设计中创建一个合适的版式分级，你不必把自己仅仅限制在字体大小上。你还可以使用颜色、字体效果（比如段首字母下沉、大写字母、小型大写字母）、斜体和其他的选择，来为用户提供关于页面内容结构和文本块相对重量的线索。

3.8 段落样式

段落是标记，是关于思想的标记。版式设计师在文本流中，使用单线边界（single-line boundaries）、缩进（indents）、段标志（pilcrows）和其他装饰，以及减少缩进、花饰字母（如首字母下沉）等技巧来标记段落。

如果你使用缩进，通常的建议是，跟随标题或副标题段落不要缩进。根据惯例，列表和引用内容块也不需要缩进。

单线边界是现在网站上最常用的段落分隔符，也是大多数通用浏览器默认设置的样式。总体来说，缩进仍然是印刷品和出版物中最流行的段落分隔符。区块和缩进的一些规约，以某种方式见证了网页和印刷品的区别。印刷的成本仍然高居不下，但是网站的成本却不再是什么大问题。毕竟，可用性是网站版式优秀与否的唯一衡量标准。

你可以通过使用邻近同级节点选择符（adjacent sibling selector）达到这个效果，而不需要使用大量的标记。例如，如果你已经为段落设定了缩进。

```
p { text-indent: 2.5em; }
```

这时，为了移除那些包含了 h1 到 h3 标题的段落的缩进，你可以这样设置。

```
h1 + p, h2 + p, h3 + p { text-indent: 0; }
```

需要注意的是，只有在引用内容块和缩进被设置成和外挂标点（hanging punctuation）左对齐时，才需要这样做。Robert Bringhurst 建议，“如果段落的缩进是适度的，考虑到连贯问题，你可以让引文使用相同的缩进。”这对网站上的列表也适用。在这两种情况下，需要用边界把列表和引用内容块从周围的段落中分离出来。

所有的浏览器都能很好地支持基本的段落样式。然而复杂的花饰字母和章首可能会引起麻烦。一些浏览器对诸如伪元素（pseudo-elements）和邻近同级

节点选择符这类技术的支持还不成熟。开发者为正文文本选择不同字体的能力也同样受到限制，同时跨平台和跨浏览器的不连贯性也使创意和精确度受到影响。

4. One em indent with boundary

Much of the material for this volume was collected during the time that I was preparing for the press the Evolution of Woman, or while searching for data bearing on the subject of sex-specialization. While preparing that book for publication, it was my intention to include within it this branch of my investigation, but wishing to obtain certain facts relative to the foundations of religious belief and worship which were not accessible at that time, and knowing that considerable labor and patience would be required in securing these facts, I decided to publish the first part of the work, withholding for the time being that portion of it pertaining especially to the development of the God-idea.

As mankind construct their own gods, or as the prevailing ideas of the unknowable reflect the inner consciousness of human beings, a trustworthy history of the growth of religions must correspond to the processes involved in the mental, moral, and social development of the individual and the nation.

By means of data brought forward in these later times relative to the growth of the God-idea, it is observed that an independent chain of evidence has been produced in support of the facts recently set forth bearing upon the development of the two diverging lines of sexual demarcation. In other words, it has been found that sex is the fundamental fact not only in the operations of Nature but in the construction of a god.

In the Evolution of Woman it has been shown that the peculiar inheritance of the two sexes, female and male, is the result of the bias given to these separate lines of development during the earliest periods of sex-differentiation; and, as this division of labor was a necessary step in the evolutionary processes, the rate of progress depended outdent-caply on the subsequent adjustment of these two primary elements or forces. A comprehensive study of prehistoric records shows that in an earlier age of existence upon the earth, at a time when woman's influence was in the ascendancy over that of man, human energy was directed by the altruistic characters which originated in and have been transmitted through the female; but after the decline of woman's power, all human institutions, customs, forms, and habits of thought are seen to reflect the egoistic qualities acquired by the male.

英文段落的样式不必一致。在这个例子中，第一段的第一行左对齐，后面段落的第一行有 1em 的缩进，右侧未对齐，形成了 V 型边界。

10. Drop cap (versal) with bold opening

Much of the material for this volume was collected during the time that I was preparing for the press the Evolution of Woman, or while searching for data bearing on the subject of sex-specialization. While preparing that book for publication, it was my intention to include within it this branch of my investigation, but wishing to obtain certain facts relative to the foundations of religious belief and worship which were not accessible at that time, and knowing that considerable labor and patience would be required in securing these facts, I decided to publish the first part of the work, withholding for the time being that portion of it pertaining especially to the development of the God-idea.

As mankind construct their own gods, or as the prevailing ideas of the unknowable reflect the inner consciousness of human beings, a trustworthy history of the growth of religions must correspond to the processes involved in the mental, moral, and social development of the individual and the nation.

By means of data brought forward in these later times relative to the growth of the God-idea, it is observed that an independent chain of evidence has been produced in support of the facts recently set forth bearing upon the development of the two diverging lines of sexual demarcation. In other words, it has been found that sex is the fundamental fact not only in the operations of Nature but in the construction of a god.

In the Evolution of Woman it has been shown that the peculiar inheritance of the two sexes, female and male, is the result of the bias given to these separate lines of development during the earliest periods of sex-differentiation; and, as this division of labor was a necessary step in the evolutionary processes, the rate of progress depended outdent-caply on the subsequent adjustment of these two primary elements or forces. A comprehensive study of prehistoric records shows that in an earlier age of existence upon the earth, at a time when woman's influence was in the ascendancy over that of man, human energy was directed by the altruistic characters which originated in and have been transmitted through the female; but after the decline of woman's power, all human institutions, customs, forms, and habits of thought are seen to reflect the egoistic qualities acquired by the male.

此处首字母下沉的处理方式没有任何垂直校正，段落也没有对齐，没有边界。

在“网站排版和设计[1]”这篇文章中，Jon Tan 对你可以在设计中使用的样式技巧和 CSS 代码进行了介绍。

[1] http://jontangerine.com/silo/typography/p/。

11. Outdent cap (versal) with bold small caps opening

M**UCH OF THE MATERIAL FOR THIS VOLUME WAS COLLECTED DURING THE** time that I was preparing for the press the Evolution of Woman, or while searching for data bearing on the subject of sex-specialization. While preparing that book for publication, it was my intention to include within it this branch of my investigation, but wishing to obtain certain facts relative to the foundations of religious belief and worship which were not accessible at that time, and knowing that considerable labor and patience would be required in securing these facts, I decided to publish the first part of the work, withholding for the time being that portion of it pertaining especially to the development of the God-idea.

As mankind construct their own gods, or as the prevailing ideas of the unknowable reflect the inner consciousness of human beings, a trustworthy history of the growth of religions must correspond to the processes involved in the mental, moral, and social development of the individual and the nation.

By means of data brought forward in these later times relative to the growth of the God-idea, it is observed that an independent chain of evidence has been produced in support of the facts recently set forth bearing upon the development of the two diverging lines of sexual demarcation. In other words, it has been found that sex is the fundamental fact not only in the operations of Nature but in the construction of a god.

In the Evolution of Woman it has been shown that the peculiar inheritance of the two sexes, female and male, is the result of the bias given to these separate lines of development during the earliest periods of sex-differentiation; and, as this division of labor was a necessary step in the evolutionary processes, the rate of progress depended outdent-caply on the subsequent adjustment of these two primary elements or forces. A comprehensive study of prehistoric records shows that in an earlier age of existence upon the earth, at a time when woman's influence was in the ascendancy over that of man, human energy was directed by the altruistic characters which originated in and have been transmitted through the female; but after the decline of woman's power, all human institutions, customs, forms, and habits of thought are seen to reflect the egoistic qualities acquired by the male.

网站核心字体预览。

3.9 通过字体设定版式

那些想创建一个丰富的、真正跨浏览器的、一致的版式设计的设计师所面对的诸多限制，是现代网站排版中最让人恼火的地方。因为操作系统预装的字体不同，很难预言某种特殊的字体是否会在用户的显示器上正常显示，或是否会产生排版噪音（typographic noise），导致文本难以阅读、布局难以浏览。

当然，字体选择必须被限制在所谓的“网站主流字体”（Core Web fonts）中，这个集合由 10 种在 1990~2000 年间设计的字体组成，作为网站的标准字体包。现在，这些字体被默认安装在了全世界 95% 以上的电脑上，因此经常作为设计师为正文选择字体的首选。

Andale Mono by Steve Matteson
Arial by Robin Nicolas and Patricia Saunders
Arial Black by Robin Nicholas and Patricia Saunders
Comic Sans MS by Vincent Connare
Courier New by Adrian Frutiger and Howard Kettler
Georgia by Matthew Carter
Impact by Geoffrey Lee
Times New Roman by Stanley Morison with Starling Burgess and Victor Lardent
Trebuchet MS by Vincent Connare
Verdana by Matthew Carter

网站核心字体预览。

这些字体是 Andale Mono (monospaced sans serif typeface)、Arial (sans serif)、Comic Sans MS (script)、Courier New (monospaced slab serif)、Georgia(serif)、Impact (sans serif)、Times New Roman (serif)、Trebuchet MS (humanist sans serif)、Verdana (humanist sans serif) 和 Webdings (dingbat)。

近些年来，Palatino (Mac, old-style serif)、Helvetica (sans serif)、Lucida、Sans Unicode (sans serif)、Tahoma (humanist sans serif) 和 Lucida Grande(humanist sans serif) 也比较流行。

Georgia、Verdana、certain Lucida fonts、Monaco、Trebuchet 和 Verdana 字体是为屏幕设计的，目的是为了提高网站文本的清晰度。和声名显赫的 Times New Roman 一起，Arial、Lucida、Georgia 和 Trebuchet 等字体在 26 像素或更大像素时效果很好，很适合作为正文的标题。Verdana、Georgia、Lucida 和 Arial 通常是文本块使用的字体。然而，在大多数情况下仔细考虑文本的行高和行宽是非常必要的。仅就字体而言，它还不是终极解决方案。

随着 Windows Vista 和 Microsoft Office 2007 的到来，微软引入了 ClearType Font Collection 2007，在包含了 6 种预装在电脑中的新字体，它们是 Calibri、Cambria、Candara、Consolas、Constantia 和 Corbel。虽然并没有被大多数人使用，它们却被预装在很多基于 Windows 系统的电脑上，也可以被安装在苹果电脑上。

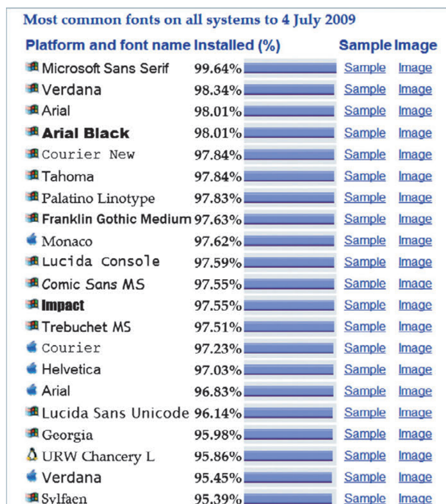
很多软件包会自动预装一些字体。比如 Office 2003 会安装 100 种额外的字体。当然，不是所有的这些字体都特别精致，也不是所有的这些字体都适合在网站上使用。然而它们仍然增加了你的选择。你也许希望检查一下 Richard Rutter 的“字体矩阵”（Font Matrix [1]），它列出了不同版本 Mac 和 Windows、Windows Office 和 Adobe Creative Suite 中打包的字体。

3.10　CSS 字体库（CSS Font Stacks）

虽然安全字体的选择受到了限制，你不必仅仅使用那些样式表里的字体。CSS 字体库里的字体，是一个按优先顺序排列的 CSS 所钟爱的字体的列表，你可以告诉用户浏览器在这个列表中循环，直到找到一个安装在用户操作系

1　http://24ways.org/2007/increase-your-font-stacks-with-font-matrix。

统中的字体，然后使用它显示任何被定义的文本。



Most common fonts on all systems to 4 July 2009

Platform and font name	Installed (%)	Sample	Image
Microsoft Sans Serif	99.64%	Sample	Image
Verdana	98.34%	Sample	Image
Arial	98.01%	Sample	Image
Arial Black	98.01%	Sample	Image
Courier New	97.84%	Sample	Image
Tahoma	97.84%	Sample	Image
Palatino Linotype	97.83%	Sample	Image
Franklin Gothic Medium	97.63%	Sample	Image
Monaco	97.62%	Sample	Image
Lucida Console	97.59%	Sample	Image
Comic Sans MS	97.55%	Sample	Image
Impact	97.55%	Sample	Image
Trebuchet MS	97.51%	Sample	Image
Courier	97.23%	Sample	Image
Helvetica	97.03%	Sample	Image
Arial	96.83%	Sample	Image
Lucida Sans Unicode	96.14%	Sample	Image
Georgia	95.98%	Sample	Image
URW Chancery L	95.86%	Sample	Image
Verdana	95.45%	Sample	Image
Sylfaen	95.39%	Sample	Image

所有系统最常用的字体，来自 Codestyle.org 于 2009 年 11 月公布的“联合字体调查结果”（Combined font survery results）[1]。

例如，如果你想用 Baskerville 字体，可以把它和作为替换的字体一起列在 CSS 字体库中。

```
body { font-family: Baskerville, Times New Roman, Times,
    serif;
}
```

如果用户的电脑上没有安装 Baskerville 字体，浏览器会检查下一个字体，也就是 Times New Roman 字体，然后是 Times，如果这些字体都没安装，浏览器将使用任意一个衬线（serif）字体。

正如 Nathan Ford 在它的文章“优化 CSS 字体库”（Better CSS Font Stacks [2]）中提到的，当你选择字体库的时候，你需要首先考虑文本的上下文。一些字体显示正文的效果较好，然而另一些显示标题的效果较好。考虑一下如果文

[1] http://www.codestyle.org/css/font-family/sampler-CombinedResults.shtml。

[2] http://unitinteractive.com/blog/2008/06/26/better-css-font-stacks/。

本元素是标题或段首的情况。总体来说，字库应该遵循以下的方案。

字体名称：理想字体、替代字体、一般字体、通用字体。

作为标题最理想的字体，你可以在“安全”字体外探索排版的可能性。很多字体具有良好的市场渗透力，所以除非你选择了一个非常不常用的字体，总有一些用户会喜欢它。对于正文字体的“理想”程度，可以通过它适合整个设计的程度和易读性来衡量。

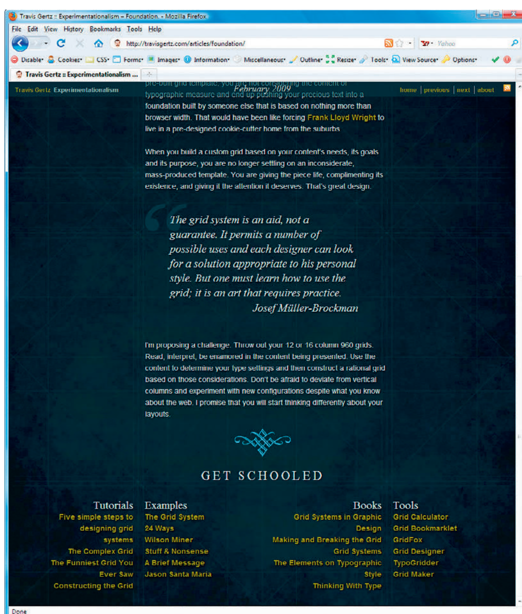
标题和文本块的“替换字体”通常和“理想字体”相似（比如字体形状），但是比“理想字体”更通用，安装得也更广泛。

“一般字体”和前两者类似，但是不具备复杂的特性，通常比较简单也比较普通。

最后，“通用字体”是在前面所有字体都没有被安装的情况下的选择，它将以任意通用的字体显示内容。

下面是标题使用的主流字库。

- Baskerville、“Palatino Linotype”、Times、Times New Roman（衬线）
- Cambria、Georgia、Times、Times New Roman（衬线）
- Franklin Gothic Medium、Arial Narrow Bold、Arial（无衬线）
- Futura、Century Gothic、AppleGothic（无衬线）
- Garamond、Hoefler Text、Palatino、Palatino Linotype（衬线）
- Geneva、Verdana、Lucida Sans、Lucida Grande、Lucida Sans Unicode（无衬线）
- Georgia、Times、Times New Roman（衬线）
- GillSans、Trebuchet、Calibri（无衬线）
- Helvetica、Helvetica Neue、Arial（无衬线）
- Lucida Sans、Lucida Grande、Lucida Sans Unicode（无衬线）
- Palatino、Palatino Linotype、Hoefler Text、Times、Times New Roman（衬线）。



在试验中，Travis Gertz 使用了不同的 CSS 字库，来为用户提供美丽的版式设计。在上面的例子中，使用了 Helvetica Neue 作为正文的字体，Baskerville 作为日期、标题和引用文本块的字体。

- Trebuchet、Tahoma、Arial（无衬线）
- Verdana、Tahoma、Geneva（无衬线）

下面是正文使用的流行字库。

- Arial、Helvetica Neue、Helvetica（无衬线）
- Baskerville、Georgia、Cambria、Times、Times New Roman（衬线）
- Cambria、Georgia、Times、Times New Roman（衬线）
- Century Gothic、Apple Gothic（无衬线）
- Consolas、Lucida Console、Monaco、monospace（无衬线）
- Courier New、Courier、monospace（衬线）
- Futura、Century Gothic、AppleGothic（无衬线）

- Geneva、Lucida Sans、Lucida Grande、Lucida Sans Unicode、Verdana（无衬线）

在他的试验中，Travis Gertz 使用了不同的 CSS 字库，来为用户提供美丽的版式设计。在上面的例子中，他使用了 Helvetica Neue 作为正文的字体，Baskerville 作为日期、标题和引用文本块的字体。

仔细考虑字体问题可以预先避免很多问题。

当你定义字体库时，要确保你选择的字体不会太罕见，以至于在整个设计中替换字体和一般字体也无法正常匹配。把非常不同的字体放在一个字体库中通常不是一个很好的主意，因为字体大小、行高和行宽的不同可能会使正文和标题产生令人吃惊的排版效果，甚至使它们变得不易阅读或使版面零乱。最后，你需要考虑一下各种可能的情况，比如，你的标题字体库中第二个字体被使用，而你的正文字体库中第三个字体被使用。在这种情况下文本是否是连贯可读的？其他的组合情况下又会是什么样？

仔细考虑字体问题可以预先避免很多问题。

3.11　文本替换技术

虽然“核心网站字体”可以产生连贯的、平台独立的版式，一些设计师在提供一些非常规的设计方案时，倾向于使用不常见的字体。不同的文本替换技术可以把纯文本替换成嵌入了所选字体的图片或动画，并把它展示给用户——特别是当客户电脑中并没有这些字体的时候。这些技术通常依赖于客户端的 JavaScript 支持或服务器端的脚本来现场生成嵌入字体的图片文件，同时它们确保内容对大多数用户而言是容易获取并便于阅读的。

总体来说，这里有 4 类文本替换方案。

- **静态图片替换：**这种技术设计师仅仅把要替换的文本替换成一个嵌入文中的图片，同时保持内容对 SEO 的友好性。Radu、Phark 和 Malarkey 图片替换法推荐使用大的外边距、字母空隙和文本缩进，相应地，隐藏文字并使用背景属性来显示替代图片。注意，为文本内容（不使用上文描述的图片替换技术）使用一个 img 元素会破坏基于语义的

HTML 标记，因此应该避免。

- **动态图片替换：**这种技术超越了静态图片替换，能够通过 JavaScript（客户端）或 PHP（服务器端）自动生成图片。设计师把字体文件上传到服务器上，服务器上的脚本将使用它生成图片并替换文本。使用这种方法的技术包括 Scalable Inline Image、Replacement、Dynamic Text Replacement、Facelift Image Replacement 和 Scalable Jens Image Replacement。
- **动态影片替换：**这种技术依赖于包含嵌入字体的 Flash 影片。这些影片在文本需要替换的时候会播放出来（比如使用，sIFR）。这种技术的优点在于，文本可以被选择并被复制。然而，这种技术增加了页面载入的时间，并且要求用户浏览器必须同时支持 JavaScript 和 Flash。如果不考虑这些使用上的严重局限，sIFR 可能是最受欢迎的网站文本替换技术，因为，它有大量的插件和第三方应用程序。

使用中的 sIFR（Scalable Inman Flash Replacement）：Meta Pro Book 嵌入了一个 Flash 影片。

- **动态画布或 VML 替换：**这种技术包含两个独立的部分，字体生成器，它会通过使用 VML 和转换引擎（比如 Cufón[1]，是一种不需要任何插件，具有跨浏览器的兼容性，并且快得无与伦比的技术）把字体转化成专用格式的文件。这种技术的优势是用户可以选择并复制整个页面的文本，而不必特别选择被替换的文本，sIFR 和其他技术是做不到这一点的。另一个例子是 Typeface.js[2]。

不管你选择什么技术，即使用户的电脑中没有安装 Flash 和 JavaScript 或在他们的浏览器中不可用，也要确保可以获取内容并且可读。静态图片替换的主要问题是获取性问题。动态替换技术通常会增加服务器载荷，所以有的时

[1] http://wiki.github.com/sorccu/cufon。

[2] http://typeface.neocracy.org。

候需要一些缓存。Cufón 和 sIFR 仍然是最近越来越流行的技术；Cufón 通常要比 sIFR 更受欢迎，因为它的跨平台兼容性，并且它属于轻量级技术，虽然它的配置不像 sIFR 那样灵活。



Dave Shea 对 Cufón 的使用：正文更加雅致，阅读起来更加舒服。

3.12　@font-face 属性

使用文本替换技术嵌入字体经常很费时并且显得不自然，因为设计师必须把字体嵌入到 Flash 中，或者使用服务器端的类库生成嵌入版面的图片。这些技术的不利方面是，它们通常依赖浏览器对 JavaScript 的支持，这在实际中很难控制。很显然，使用纯 CSS 对本地字体的支持将会是一个更可取、更自然的解决方案。这正是 CSS3 网站字体模块 @font-face 属性所支持的。

使用 @font-face 嵌入字体要通过两步完成。第一步是把一个新的字体加入样式表中的可用字体列表。这通过使用 @font-face 属性完成。

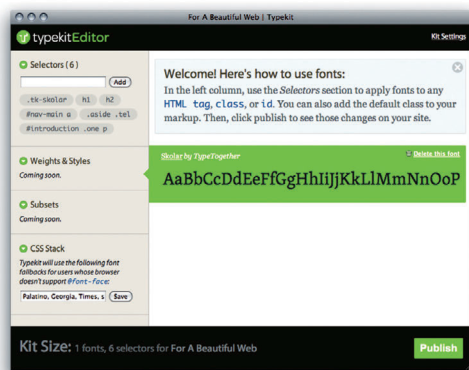
```
@font-face {
font-family: MuseoSans;
src: url('fonts/museo_sans.otf') format ("opentype");
```

```
font-weight: bold;
}
```

在这个例子中，设计师声明 Museo Sans 是一种用来在浏览器中显示文本的可用的字体。接下来的第二步，是设定一个文本元素，使用刚才声明的字体，我们在常规的 CSS 声明中使用字体，如下所示。

```
h1 {
font-family: MuseoSans, Arial, sans-serif;
}
```

浏览器将会载入 museo_sans.otf 字体文件，并用它来显示 h1 标题。如果浏览器不支持 @font-face 属性，它就会被忽略。要注意只有可以免费获取的字体才会被上传到服务器，因为这些文件不仅仅对浏览器开放，也对用户开放，用户可以从服务器下载这些字体文件。字体的使用协议需要清晰地表述，该字体可以被使用在 @font-face 属性中。



Typekit 是一种简单的解决方案，它提供仅限于网站使用的字体使用协议的链接，允许你在布局中嵌入商业字体（图片由 Andy Clarke 提供）。

还可以选择 Typekit [1]或 Fontdeck [2]，这两个服务提供了一种简单的解决方案，

[1] http://blog.typekit.com/2009/05/27/introducing-typekit/。

[2] http://ilovetypography.com/2009/08/07/the-font-as-service/ 。

它们通过提供仅限于网站使用的字体使用协议的链接，允许你在布局中嵌入商业字体。字体文件，或者某种字体不同重量的字形，被存储在第三方的服务器上，设计者向 Typekit 提交租用（而非购买）协议。关于付费，设计者会被给予一个包含在页面上的 JavaScript 文件，它可以和简单的 CSS 一同使用。

3.13　选择正确的字体

超过一千种字体可供选择，要为某个特殊的目的选择正确的字体似乎是一个让人望而生畏的工作。最常见的错误是选择漂亮的字体，一些字体看上去很有吸引力，人们因为形式而忽略了功能。这相当于本末倒置。虽然这听起来有点奇怪，但是字体的“外观”应该是你最后考虑的事情。但是，如果外观不是那么重要，那我们的决定该以什么作为依据呢？最重要的标准是字体的上下文，以及使用字体的目的。

考虑上下文和故事本身

在开始选择字体和布局的细节性操作之前，如果有可能要通读一下文本。字体的选择和它的展示方式要由文本的意义决定，这似乎是显而易见的。然而，这也有一个显而易见的问题。在互联网时代，我们经常要处理两类不同的文本内容。

规定性内容：这是我们知道并且喜爱的一类内容。设计师在开始设计之前已经获悉文本的内容和含义。读者可以从头到尾把内容读完，并在阅读过程中获得启发。

突发性内容：这类内容在设计完成后才会产生。想一下社交网络，每个人都参与内容的构建，或者是包含 CMS 或 Blog 引擎的网站，在它们会不断产生新的内容。读者可能会快速浏览内容，在页面内容间进行跳跃，并在这个过程中收集有用信息，在那些吸引他们注意力的地方停下目光。

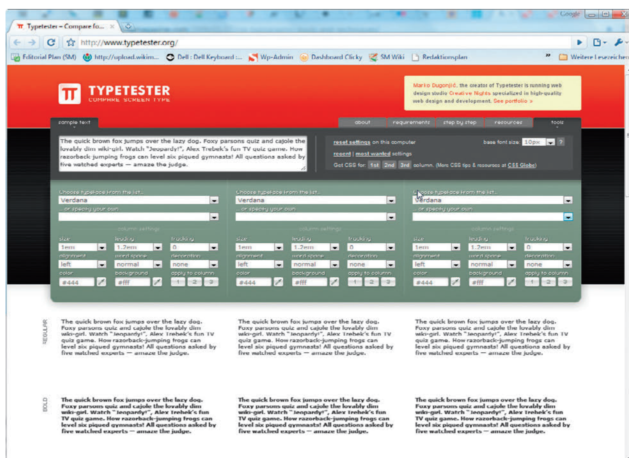
对于规定性内容，我们可以提前阅读，理解它的含义、语气和微妙之处，我们可以以此为根据选择相应的字体。但是对于那些突发性内容该怎么办呢？当然，我们可以尽我们的所能。看一下品牌和环境。设想一下文字出现的上下文。选择一种能够丰富内容的字体，它最好能够融入背景而不是为了吸引注意力而显得突兀。

3.14 不要穿舞鞋登山

实际上每个字体设计师都是按严格定义的标准和特殊的功能来创造一种字体。而网站设计师应该在选择字体之前认识到它的功能。在字体系列的一端是可以以任何大小显示、并适用于任何恶劣条件的高负荷字体（登山鞋），在另一端是展示字体（舞鞋）。

高负荷字体有稳健的特性。通常容易辨识、较大的字体空间、固定的衬线、清晰的对比、较高的 x-height 和墨圈（边角处防止墨水外溢的密闭的区域）。墨水外溢问题出现在网站上不算什么问题，但是其他的因素却在起作用。展示字体更加雅致，字形上有更多的选择。这是一个处于很多对立的极端之间的世界，很多位于两级的字体是出于特殊的目的而被设计出来的。处于中间的字体非常重要，它们既适用于常规的文本，也适合标题。

高负荷字体比展示字体更灵活。在某种程度上，一个人可以穿登山鞋参加盛宴。如果你配上洒脱的牛仔服、时髦的衬衫和夹克，这并不是什么问题，这就像高负荷字体作为标题看上去也可以接受，前提你调整它们的空隙，并把它们和正确的设计元素一同使用。但是，你不应该穿舞鞋去登山。当然，你可以，但是你这样做了之后不要哭。同样，永远不要在正文中使用展示字体。



Marko Dugonjic 的 Typetester 是一个流行的在线应用程序，它允许你测试不同的字体、行高和行宽在显示器上的显示效果，并允许你选择最好的一个。

3.15　尺寸很重要

当然，上述的很多字体必须处理尺寸问题。随着可拉伸数字化字体的到来，任何字体的任何尺寸都可以使用，以往耗时的字号调整过程已经被抛弃了。后果是很多设计因为这种便利而受害。一些展示字体在小号时太过于华丽或脆弱，用来显示文本的字体在被放大后显得过于笨拙。



虽然它们看起来相似，文本字体和展示字体在放大后显示出了很多不同。

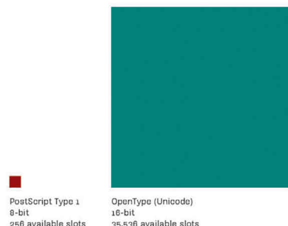
字体大小的优化不仅对印刷品很重要。当为显示器进行设计时，类似的问题也会出现。笔画的浓度和像素的大小是关键的。当字体太小、线宽低于一个像素的时候，精致的字形将会断裂，衬线的边际将会消失。如果字体的主体没有占据足够多的像素，x-height 较低的字体将变得难以阅读，会产生不合规范的或复杂的图形、关闭的围白和排列过紧的字母。

3.16　OpenType、TrueType 还是 PostScript Type？

现在数字化的字体有 3 种可供选择的格式：OpenType、PostScript Type 1 和 TrueType。这可能有一点混乱。了解文件的格式是否和你将要使用的软件兼容是非常重要的事情。

总体来说，如果你在使用一个现代图形编辑软件，使用 Opentype 格式的字体几乎不会遇到任何问题。两个主要的优点是跨平台兼容性（你可以在 Mac、Windows 和其他操作系统上使用相同的字体）、支持扩展字符集和布

局特征。它们提供丰富的语言支持和高级的排版控制。现在，所有 Adobe 的应用程序（InDesign、Illustrator、Photoshop 等）都提供对 OpenType 的高级支持，比如自动替换替代字形、自动连字、小型大写字母和 swaches 字形和古旧风格（old-style）特征。



一个 PostScript Type 1 字体可以包括 256 种字形；一个 OpenType 字体文件可以包含 2562 或 65536 种字形。

注意，OpenType 字体可以和 PostScript Type 1 和 TrueType 字体一同安装和使用。因为这些字体基于 OpenType 特有的表格，那些在 Mac OS X 和 Windows 2000 以前的系统上不兼容的应用程序在没有 ATM 系统服务的前提下是无法使用它们的。即使在最新的电脑和最新的操作系统上，一些程序仍然是落后的。

3.16.1　你在追求什么：家常便饭还是美味大餐?

字体不仅仅只包含你在键盘上看到的那些符号。除了字母表、数字和各类标点之外，大量字体也包含一系列“隐藏的”字符。它们可以通过 Shift 和（或）选择键或通过特殊的菜单获取。标准字符集里面大概有 250 种字形，它们足够用来满足显示和基本文本使用的需要。然而，正如在一家高档餐厅中你不可能只点汉堡和薯条，好的版面需要的会更多一点。如果你正考虑为某种专业的版面选择字体，确保可以保证以下这些额外的特征。

比如，小型大写字体（也被称为 small caps）要比 x-height 高一点。它们在首字母缩写和避免把全部字体都变成大写时使用，这会在文本中非常显眼。它们也可以用来突出内容或作为章节、段落的首字母或首行。微型大写字母会更小一些，和 x-height 的高度差不多。一些设计师使用它们来进行排版试验，比如进行混合大小写设置（同时包含大写和小写），也被称为统一大小写设置。

标准的字体通常有一个字符集，然而不同的排版会要求不同的风格。符合比例的突出字符会和运行文本很好地融合；符合比例的衬垫字符适合任意大小写的文本；平坦的字符适合设置表格和数字计算器；上标和下标字符适合用在分数和科学文本中。最后一类文本不仅需要上标字符和下标字符，它还需要特殊的数学符号和很多种类的括号。

当选择一个合适的字体时，要确保字体是完整的，同时包括你在设计中需要的全部字形。比如，一些廉价的字体可能不包括德语的元音符号（ä、ö、ü）和 ß 字符。

实际上，当字符主体的一部分伸入到下一个字符的范围内时，就有必要使用连字。很多数字化字体仅仅包括一种受限的“f”连字，这是不够的。一个扩展的连字确保了任何扩展的连字字符都会有优雅的外形。一些字体设计包含了非常不常用的连字，经常会添加一些 swoosh 字符到字体中。swoosh 字符可以被添加到 swash 字体中，作为大写首字母、首字母或尾字母。

在购买字体之前要仔细地看看字体。分析一下这个字体的家族，仔细考虑它的字形在放大的时候的样子（100~150 点）。也要看一下字体家族不同的成员在你通常使用的尺寸下的外观。它们的斜体容易阅读么？小型大写字母足够清晰么？黑体和斜体放在一块效果如何？向“rococo”这样的词是否容易阅读？在单词“l0lo”中是否可以区分 0（零）和 o（小写字母 o）？在“floria”和“Eigenschaft”中的连字情况如何？



专业的排版需要呈现大型小写字母、很多的数字集、扩展连字集、不同种类的专业字形，包含替代字符和 swash 字符也无大碍。上图是 Capsa 专家字体。

3.16.2 衬线还是无衬线？…… 或者 monospace 或 cursive？

时下正在进行的关于衬线和无衬线的争论不得不涉及清晰度问题。双方都有言辞激烈的支持者。随着屏幕尺寸、分辨率和像素密度的增加，关于毛边较少的无衬线字体会增加清晰度的论据失去了一些优势。像 Host Hochuli 在《排版细节》（Detail in typography）一书中展示的那样，衬线字体可能更容易阅读，而且很多人对它们的熟悉程度，以及对网上通常使用的字体的依恋，不管怎

样都会使这些字体更容易阅读。即使专门为显示器设计的字体在跨平台方面也会遇到困难。在一台显示器清晰易读的字体在另一台显示器上可能会有所不同。

考虑到清晰度，在为显示器选择字体时这里有一些关键的参考因素。

熟悉的字母样式。选择大家熟悉的样式。避免使用那些打破常规的“奇异”字体。永远不要把正文全部设置成大写或全部使用小型大写字母，要注意斜体字（italic）、倾斜字（oblique）和其他变体。比如，升部和降部对清晰度非常重要。在选择字体时，要选择“b”、“d”这类字母的升部高于 x-height，以及“p、“q”这类字母的降部低于 x-height 的字体。

选择为显示器设计的字体：很多字体并非为显示器设计，甚至在设计时没有进行过优化。如果一个字体的线条没有被很好地设计，显示时就会出现问题。比如，当字号很小时，大写字母“A”中间的横杠就会消失，其他衬线和精致的字形也会消失。

选择真实的字体而不是合成的字体：仔细考虑一下那些你想用的字体的可用字形。很多通用的字体仅仅有正体、粗体、斜体和粗体斜体可用。不管提供什么样的风格变体，都需要专门的设计。

浏览器可以合成字体：一个典型的例子就是合成斜体字，更确切地说是倾斜字（在不改变字母形状的前提下，把正体倾斜一定角度，来模仿真正的斜体字）。那个苹果电脑上无处不在的、由 Charles Bigelow 和 Kris Holmes 设计的 Lucida Grande 字体根本没有斜体样式。然而，这并不会阻止人们通过 CSS 展现出斜体样式。浏览器会合成它，但是看起来很糟糕。小型大写字母是另一个例子，没有一个核心的网站字体包含它。然而，我们有时会采用合成的方式。技巧是避免使用合成字体，根据你的判断力来判断什么时候这样做和为什么这样做。

充裕的 x-height。这是与整个字体高度相比，基线和小写字母“x”（由此命名）顶端的距离。一个充裕的 x-height 对于清晰度是很重要的，尤其是对显示器而言。Georgia 和 Verdana 是专门为显示器设计的字体，它们都有较充裕的 x-height。

让人舒服的字母间距：可以通过 CSS 的 letter-spacing 属性调整字母间距；一个已经具有良好间距的字体是一个非常好的出发点。

让人舒服的单词间距：这是单词间的空隙。它可以通过 CSS 调整，但是即使不调整也应该看起来很舒服。

3.16.3 美丽的字体就是它想要的东西

在全部其他条件都满足的情况下，最后可以选择你认为视觉效果最好的字体。但是你现在知道，这实际上是整个选择过程中的最后一步。理解前面的步骤的重要性是至关重要的。选择一个真正满足要求的字体要比一个仅仅看起来漂亮的字体更有意义。你要传达的消息将可能会被误解，除非字体在你的操作系统上表现良好、适用于所有需要展示的语言、有完整的字符集、是一个大的字体家族中的一员、能够正确地激发氛围和符合文化内涵。

3.16.4 互补将令你无所不能

合成字体是一门艺术。如果没有足够好的理由，请尽量少使用它。很多设计师反复使用相同的字体，依赖于每种字体中不同的变体变换风格。甚至更多人仅仅依靠几种字体，而不管他们下载了多少字体，因为这几种字体是经过考验的可信的朋友。这对网站也适用。

在不远的将来，我们可能会有成千上万种可供使用的字体，但是使用那些人见人爱和被广泛理解的字体家族，可以避免犯错误。

把衬线字体和无衬线字体混合使用是一种很好的方式。在正文中混合使用它们的时候，主要要匹配 x-height。

Georgia Verdana

由 Matthew Carter 设计的 Georgia 和 Verdana 字体，在相同尺寸时的显示效果。Verdana 似乎大一些。

Georgia Verdana

George 和尺寸略小的 Verdana 字体。它们的 x-height 相匹配，看起来似乎一样大。

就像色相环上的互补色一样，字体也存在互补关系。然而，要小心处理。某些特定字体间的对比可能会像蓝色和黄色的对比那样刺眼。

要记住，电脑显示器会产生比一般印刷品更大的对比度。因此，很多设计者倾向于在白色背景上使用灰黑色（off-black）而不是纯黑色（pure black）[1]。同样，在黑色背景上使用亮灰色（light gray）代替纯白色（pure white）也会产生更好的视觉效果。为了通过 CSS 获得高质量的排版效果，"所有的视觉区别要尽可能地细微，但是仍然要清晰而且显著"[2]。

尝试同种字体不同的风格是一个出发点。在标题中使用粗体字，在副标题中使用小型大写字母或大写字母，下一级副标题中使用斜体字，在正文中使用正体字。通过实验确定正确的页面元素的分级。

如果你没有对你的字库进行大量实验，要注意字体的基线在不同的平台的变化。你可能以为 Helvetica Neue 和 Arial 字体会有相同的基线，但实际上并非如此。在 Windows 系统上使用 Arial 字体看一个页面和在 Mac OS X 系统上使用 Helvetica 字体看一个页面的不同之处，会使网格的对齐显得倾斜。

3.17 注意细节

现在让我们来考虑一下关键的排版理念、术语、最佳实践和实际的建议，让我们从不同视角来看一下排版。为了获得美丽的、丰满的和印象深刻的版式，需要创建格式正确、谨慎编写的内容。这就是，如果你想留下深刻的印象，你需要润色你的作品，对那些微小的排版细节给予足够的重视。

寡妇和孤儿

"寡妇"（widow）就是在段落末尾的一个单独的短行或一个单独的词。"孤儿"就是在一个分栏的开头或结尾的一个单词或短行，它被从段落的其他部分分离出来。"寡妇"和"孤儿"产生了难看的碎片，打断了读者的阅读，破坏了可读性。它们可以通过调整字体大小、行宽、行高、单词间距和字母间距

[1] Jeff Croft，"Elegant Web Typography"（优雅的网站排版）。

[2] Edward Tufte，"Visual Explanations"（视觉解释）。

或手动加入换行符来避免。

不幸的是，CSS 中没有简易的方式避免“寡妇”和“孤儿”。你可以使用 JQuery 的插件 jQWidon’t 或 Typogrify WordPress 插件，它们会在文本块最后两个词之间加入一个不换行空白（ ）[1]。

Good

Lorem ipsum dolor sit amet, consectetur lout adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. dolor sit amet antetut mauris.

Nulla ac odio. Praesent bibendum justo id posuere orci quis ligula massa vulputate

egestas massa vulputate nisl mauri. Suspendisse magna tellus, faucibus, sodales, vehicula eget.

Lorem ipsum dolor sit amet, consectetur lout adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputa nisl. Curabitur venenatis nullam.

Bad

Lorem ipsum dolor sit amet, consectetur lout adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa atheis vulputate nisl. Curabitur venenatis aerues mauris.

Nulla ac odio. Praesent bibendum justo id mauris. Suspendisse magna tellus,

dapibus sodales, vehicula eget.

Lorem ipsum dolor sit amet, consectetur lout adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis antetut.

“孤儿”和“寡妇”在排版中显得很糟糕。

3.18　对齐和连字符

在排列没有对齐的文本块时，要确保使碎片（没有对齐的边缘）平衡，不要产生任何突兀的孔洞或难看的形状。一个坏的碎片可以扰乱视线并分散用户的注意力。一个好的碎片有一个柔和的非整齐的边缘，没有一行太长或太短。没有办法通过 CSS 控制它，所以你必须通过手动调整来达到好的效果。

[1] http://blog.hamstu.com/2007/05/31/web-typography-just-got-better/。

Good

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla ac odio. Praesent bibendum justo id mauris.

Bad

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa id mauris. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla ac odio.

要确保使碎片（没有对齐的边缘）平衡，不要产生任何突兀的孔洞或难看的形状。

另一个解决方案将会自动断字和对齐。现在，虽然现代的浏览器只有非常简单的断字机制，并且不支持印刷品设计中复杂的对齐。显示环境的许多变量设置使你无法手动设置断字。比如，使用软连字符（）。

在网站上，两边对齐的文本通常比左对齐的文本更难阅读和扫读。有一些服务器端和客户端具有自动连字的解决方案，比如，phpHyphenator 和 Hyphenator。牢记，对齐的文本使用衬线字体要比使用无衬线字体效果好，在比较窄的行宽下效果最好。

3.18.1 突出

在没有打扰读者的情况下突出单词是很重要的。斜体字通常被认为是突出单词的最好的形式。其他通用的突出形式有粗体、大写、小型大写、字号、颜色和不同的字体。不管你选择什么，要尽力限制你自己只用一种形式。也就是说，把大写、粗体和斜体混用，将会产生不连贯的效果，看起来很笨拙。

这里有一些使用 CSS 进行突出的不同方式（牢记，font-variant style 仅仅在字体支持 small-caps 变量时才有效）。

```
span {

font-style: italic;
```

```
}
h1 {
font-weight: bold;
}
h2 {
text-transform: uppercase;
}
```

3.18.2 标点外伸

优秀的排版会把引号、括号、数字拉伸到文本流的外边。缩进的括号会打断文本流。通过标点外伸，文本和引用内容可以被突出并显得更加精致和易读。在你的 CSS 中，确保外伸的标点在正文文本的外边距之外。对于引号，给 CSS 的 text-indent 属性一个负值（当然，这个取决于字体的大小）。

Good

“ Lorem ipsum dolor sit amet, consectetur atse adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam luoi egestas facilisis ante. ”

Bad

“ Lorem ipsum dolor sit amet, consectetur atse adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam luoi egestas facilisis ante. ”

一个关于好的引号和不好的引号展示的例子。把引号外伸到正文文本的外边距之外，这样视觉韵律不会被打断。

```
blockquote {
text-indent: -0.8em;
font-size: 12px;
```

```
}
```

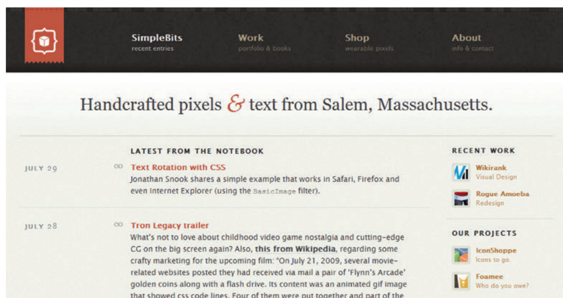
为了让有序列表（ol）和无序列表（ul）达到同样的效果，使用 list-style-position 属性，来把它们的列表符号拉到左边。几乎所有主流浏览器都支持这项技术。

```
ul, ol {
list-style-position: outside;
}
```

在人们浏览页面时，缩进的列表可以唤起人们采取行动。如果你在使用缩进，要确保你是在为某种目的而有意地使用它。

3.18.3 有格式的和号（&）

和号通常是一系列的弯曲，不同字体之间的显示效果各不相同。在 CSS 中，你可以为它选择一种特殊的字体。Simplebits 一篇名为“使用最好用的和号[1]”（Use the Best Available Ampersand）的文章，提出了通过设定可供选择的字体家族，这种有趣且有效的方式来选择最佳的和号。



在 Simplebit 的标语中有一个美丽精致的“&”。

(X)HTML：

```
<p>pixels <span class="amp">&amp;</span> text</p>
```

CSS：

[1] http://simplebits.com/notebook/2008/08/14/ampersands.html。

```
span {
font-family: Baskerville, Palatino, "Book Antiqua",
serif;
font-style: italic;
}
```

3.18.4 不要用连字符代替 em 破折号（em dash）

如果你不想打断自己，使用一个 em 破折号（—），而不要使用一对连字符（--）。这是无数的编辑最常抱怨的问题。

3.18.5 不要错误使用引号

应该“这样”使用引号（引号要像 66 和 99），而不要”这样”使用。开引号和关引号是不同的。请注意引号的选择会因语言而异。对于美国英语，引用内容通常被双引号包围，然而嵌套引号使用单引号。对于英国英语，以下两种方式都可以：先是双引号然后是单引号，或者先是单引号然后是双引号。在 CSS 中，你可以使用 :lang pseudo-class 调整引号的外形。

```
:lang(en-us)>q {
quotes: "\201c" "\201d" "\2018" "\2019" ;
}
:lang(en-gb)>q {
quotes: "\2018" "\2019" "\201c" "\201d";
}
```

现代浏览器支持这种样式，所以它被强烈推荐用来在不同地域间智能地使用[1]引号，同时避免所谓的哑引号（dumb quotes）。比如，在阿拉伯语、荷兰语和德语中引号是„这样“（看起来像 99 和 66）的，在俄语、法语和意大利语中 « 这样 » 的引号更常用。

[1] 当然即使是，IE8也不支持它，虽然它注意到了这个元素。为了避免引号编码问题，添加数字值是很重要的。

3.18.6 句子之间不要使用两倍的空格

在句子间使用两倍空格是过时方式，由于网站排版的方式最后可能会被消除。就在几年以前，要犯这个过失需要手动地插入一个空白的ASCII空白码。现在，在你允许的情况下，一些内容管理系统实际上将会格式化内容以产生双倍空格。不要给它们这个机会！在句子之间使用单倍空格。

3.18.7 必要时使用重音符

虽然重音符在HTML中很难录入，但这些非标准的字符是很重要的礼节和表示尊重的方式。很多表格列出了字符实体，包括维基百科[1]上的那个，你可以酌情参考。

3.18.8 把文本当做一种用户界面

界面上的文字选择极其重要，它可以构成或打乱网站的功能性。样式不合适的字符对于用户将要进行的交互不能提供任何指示。

作为内容的文本



作为 U1 的文本



Cameron Moll 在他那篇名为“9种技巧来区分好的设计师和伟大的设计师”（Nine skills that separate good from great designers）的文章中讨论了把文本当做用户界面来处理的重要性。[2]

以上图片“将文字作为内容”和“把文字作为用户界面”进行了对比。左侧是没有栅格化的文本，而右边的文本起到了用户界面的功能。注意文本颜色和重量的不同，以及段落间、行之间和链接之间足够的空白，它们很突出、而且容易辨识。

---

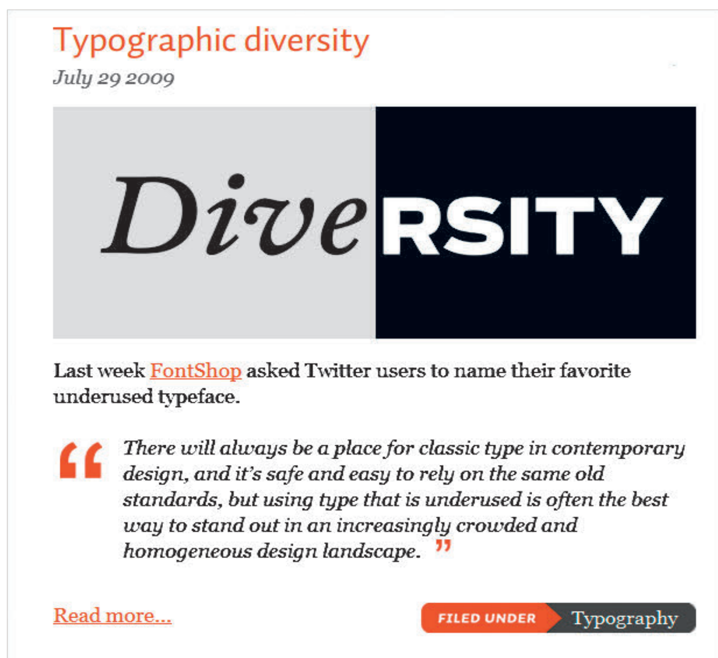
[1] http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references。

[2] http://www.cameronmoll.com/archives/001266.html。

如果你想要内容在传达意义的同时来体现功能性，以便于帮助用户理解他们该如何通过文本达到目标，你就必须包含此类微妙的视觉线索。

3.18.9　引文

引文可以突出节选的文本。它们通常用于声明之中，有时候用在博客的评论中，最重要的是它们会被用在段落文本中。然而，并不是所有的引文都是相同的。节选引文（pull quote）直接从页面自身节选文本内容，它会从内容的文本流中拉出一部分文本，并且把它们在页面显著的位置重现以吸引用户的注意力。



Idsgn.org 中一块生动的区块（block quote）引文。

正如节选引文一样，区块引文（block quote，更确切地说是 block quotations），被当做独立的段落或区块而从正文中分离出来。和节选引文不同的是，它们包含一篇不在页面内容之内、来自额外的资源的文章。区块引文通常被设定在正文文本流之内。“常规”引文引用其他资源的内容，并且和正文文

本流融合，而不是分离。

根据 HTML 的标准，有 3 个元素可以用来表示引用内容：<blockquote>、<q> 和 <cite>。虽然它们都用来标记引文，每一个都有不同的含义和不同的用途。

- 一个区块引文标记 <blockquote> 用来标记相对较长的引文，这是脱离正文的一大段内容。这些文本通常来自外部的资源，也可以参考以前的文章。
- <q> 用来表示短的内嵌的引文。

```
<p>Andrew answered <q>fonts</q>, but Steven said
<q>typefaces</q></p>
```

虽然它很少被使用，但是它有一些有用的属性。比如，你可以通过使用 CSS 在 <q> 元素的内部调整引文的外观。这是非常有用的，因为不同的语言有不同的引号形式（可以参考上文“不要使用哑的引号”那一节）。

- 最后，<cite> 定义了一种内嵌的引用或参考其他资源的格式。

```
<p>But then <cite>Andrew</cite> said <q>No, I think
these fonts work better</q>.</p>
```

引号、括号、线条、缩进、引用的图像、对话框、气泡提示——一个设计师可以采用多种方式来创建美丽的、值得记忆的引文。设计方案因颜色、形式和大小的不同而不同。不同的技术产生不同的效果。然而，有一点是很重要的，对访客而言，引文必须确实像引文，否则它将变成难以琢磨的内容。同样，节俭地使用引文，它们应该扮演配角，来支持正文内容，而没有其他的用途。它们应该有令人愉悦的变化，而不是千篇一律。

虽然看上去很漂亮，节选引文有内在的问题，它们被放在 HTML 内容的中间。

对于那些通过支持 CSS 的浏览器来浏览页面的用户来说，一切可能都非常令人满意。但是对于那些使用不支持 CSS 的浏览器或阅读器的用户而言，节选引文会出现在正文的中间。一段引文突然出现在两个段落之间，但是并没有把它们连接起来，这将会打断文本流并使用户迷惑。

如果你的内容包含节选引文，为那些误入歧途的用户提供一点额外的信息也

许是明智的选择。在 (X)HTML 中可以包含一条消息，并通过 CSS 进行隐藏，在引文的开头加上“节选引文开始”，在引文结尾加上“节选引文结束”。你甚至可以使用类似于“跳跃导航”（Skip navigation）的链接来帮助用户跳过整个节选引文继续浏览主要内容。

3.18.10　连字符和特殊符号

最后，我们将介绍一些最重要的排版符号、它们的作用和在处理高级排版和特殊的文本时的优秀实践。

3.18.11　硬连字符用来在单词中换行并连接单词

连字符（hyphen ,-）是最常用的排版符号。这个符号可以对单词进行分割和连接。连字符不是减号（–）也不是破折号（–, —, —）。软的连字符（）出现在文本行的末端，表示一个单词在下一行将会继续。“软”是指这个连字符会消失，如果整个单词正好在行末结束。“硬”是指这个连字符必须总是出现，即使在被连接的词出现在一行中（如 self-respect）。不幸的是，现在连字符还必须手动插入到 HTML 中。

硬连字符用来在单词中换行并连接单词。

en 破折号（en dash）比半个 em 破折号要长。它用来指示区间，比如在日期、时间（2:00–3:00 p.m.）、数字、比赛得分和页码中，或者用在复合形容词词组中（Meyer–Stevenson Lexicon）。在 HTML 实体中，en 破折号用“&8211;”表示。

- hyphen

– en dash

— em dash

连字符、en 破折号和 em 破折号。

一个 em 破折号有 1em 宽。它用来显示话语中的中断（I was debugging the style sheet – wait a second; what was he saying about IE6?）。在 HTML 实体中，em 破折号用“&8212;”表示。注意，在破折号周围留一些空白，比如，em 破折号可以被一个 thin space（ ）或一个 hair space（ ）包围[1]。

---

[1] 也可以看一下Jon Tan的“排版空隙设计套件”（http://jontangerine.com/silo/typography/spaces/）它包含不同类型的排版空间的一系列例子，通过使用核心网站字体在用户代理和操作系统之间进行了对比。

卷曲的单引号或双引号开始符号不是用来表示英尺或英寸的正确符号。一个单个的度量单位会被用来表示英尺或分钟（′ 在 HTML4.01 中不被支持），用一个成双的度量单位来表示英寸或秒（″ 在 HTML4.01 中不被支持）。

为了表示引语中省略的词语或者思路的减弱，使用省略号（ellipse, …）、而不是三个连续的句号（periods）。三个句号并不是省略号。这是一个常见的错误，因为它们看起来太像了，但在语义上它们是不同的。

总之，这里有一个排版中最常使用的符号的参考表格。使用数字的 Unicode 的参考是一个很好的方法，因为非数字的 Unicode 可能不能被正常解码[1]（例如，命名实体被限制在一个 DTD 中，它并不总是会进行翻译，所以一个 RSS 解码器对它进行解码时可能会产生问题）。

& 和号 (&)：通常在正规笔语、编程语言等中有很多用途。

© 版权符号 (©)

® 注册商标符号 (®)

§ 章节号 (§)：当使用两次时（§§）意味着“章节”。

’ 撇号 (')

‘ ’ “ ” 引号 (‘ ’ “ ”)

× 乘号 (×)

° 度数 (°)

¼ 四分之一 (¼)

½ (½)

¾ (¾)

‰ 千分号 (‰)

€ 欧元号 (€)

[1] http://people.w3.org/~dom/archives/2005/04/named-versus-numeric-entities/。

…　　省略号 ( )

–　　En 破折号 (–)

—　　Em 破折号 (—)：用来表示思路的突然停顿。

―　　水平号 (―)：用来引入引文。

空白符：这不是符号，什么都不是，只是一个空白空间。但是你可以用它来制造不同，尤其是在排版中。

3.19　总结

本质上，排版是一个进行简单、有效交流的有力媒介。在网上，排版可以赋予内容活力，把无生命的数据块，变成活的、优雅的对话。但是你必须仔细考虑大量的排版细节，不仅仅是字体和展现它的上下文，还有行宽、行高、字距、对比和字号。排版辅助工具，比如网格和垂直节奏为产生和谐的布局提供了一种有力的框架，通过它们，版式可以呼吸并满足它的目的。

正常的段落格式、合理地使用空白和排版分级以及缩放，可以改进文本的结构，使它更易被浏览和阅读。不仅如此，还要密切注意正文的质量，包括标点符号、空白符号和特殊符号，这有助于提升阅读体验，并对整个设计的可用性有很大的影响。因为网站排版即是与用户进行交流，你最好确保对话是丰富的、有意义的，这样用户才会欣赏它。

第 4 章

现代网站的可用性原则

Andrew Maier　David Leggett

设计一个可用性强的网站，对任何一个设计者来说都不失为一个雄心勃勃的目标。为了完成这一目标，作为有经验的设计师，我们采取的每一步都应该使我们的网站更加接近用户的预期，这是一个缓慢的过程。我们现在采用逐步递增的方式实现并测试我们的设计，直到我们达到这个整体目标。这与过去网站和网络应用的设计方式有所不同。

一般来说，在网站设计、开发和发布后，可用性专家会帮助网站的主人使网站的界面达到用户的预期。在网站发布后发现界面错误，通常是一件极其糟糕的事情。在这个阶段修复界面错误，代价可能是高昂的。因此，把可用性的操作知识铭记在心，分阶段设计、及早测试和经常测试是非常重要的。

记住，在大多数情况下，设计师的想法并不能代表用户。

- 如果一个设计师乐于向你提供反馈，你相当于在购物的同时获得了奖励积分。但是请记住，设计师的想法并不能代表用户。要与用户保持一致，我们必须不仅仅专注于网站的外观，而这很难做到。
- 为了把事情复杂化，现代的网站迅速变成了多面结构体。相互竞争的网站通常提供相似的功能，但是只有提供了最佳用户体验的那一个才会胜出。这是网站前进的方向，也是网站设计的一种进化。

创建非凡的用户体验需要一点点技巧、一点点运气和一系列细节性的工作。那些细节就是你的用户会反复留意的东西。同时，他们容易忽略用户体验领域所包含的很多相关原则，比如视觉设计、信息构建和可用性分析。在本章，这些相关的话题我们都会涉及，同时还会提供例子和实用的建议，你可以用它们来新建或优化一个网站。

4.1　用户如何思考?

用户上网的习惯与顾客逛商店的习惯基本上没有太大的不同。他们以访客的

目光浏览每个新页面，扫视某些文本，然后点击第一个让他们感兴趣的或是大体上与他们正在寻找的东西类似的链接。实际上，有大量网页他们几乎不看一眼。

大多数用户寻找有趣的（或有用的）而且可点击的元素。这样的一个元素出现，用户就会点击。如果打开的新页面没有达到他们的预期，他们就会点击“后退”按钮，并继续搜索。

- 用户重视质量和可靠性：如果他们看到高质量的内容，他们通常愿意忍受广告和糟糕的设计。这就是为什么多年来那些设计平庸而内容优良的网站会获得许多流量的原因。内容本身比用来支撑它们的设计更为重要。
- 用户不会阅读，而是浏览：当他们分析一个网页时，他们通常会寻找一些能够帮助他们略过页面其他内容的固定点或是锚点。
- 网站用户是没有耐心的：如果一个网站没有达到用户预期，那么设计就是失败的，公司也会蒙受损失。用户处理一个网站信息所需的认识负荷[1]（cognitive load）越高、导航的直观性越差，用户就越容易离开，转而寻找其他替代的网页。
- 用户并非总会做出最佳的选择：他们并非总是探寻如何以最快的方式查询想要的信息。他们也不会以一种直线方式，一部分接着另一部分地扫视页面。相反,用户会选择那些“充满”(既“充分”而又令人“满意”)的信息。他们会选择第一个合理的、表述恰如其分的选项。用户一旦发现一个可能会达到他们目标的链接，他们非常可能会立即点击它。优化是艰难的，需要很长的时间。使网站变得“充满”效率会更高一些。
- 用户遵循他们自己的直觉：在大多数情况下，用户会靠自己的摸索而不是靠阅读设计者提供的信息解决问题。根据史蒂夫·克鲁克（Steve

[1] 译者注：认知负荷理论（cognitive load theory）是Sweller等人在20世纪80年代提出的，主要从认知资源分配的角度考察学习和问题解决。Sweller等人认为，问题解决和学习过程中的各种认知加工活动均需消耗认知资源，若所有活动所需的资源总量超过个体拥有的资源总量，就会引起资源的分配不足，从而影响个体学习或问题解决的效率。

Krug）的观点[1]，一个基本原因是用户对此并不在意。“如果我们发现一些方法行得通，我们就会坚持这么做。对我们来说，只要我们能够使用它们，是否理解它们的工作原理就无关紧要。如果你的观众将按你设计广告牌的方式行事，那么你就设计出了绝佳的广告牌。”

- 用户希望能够进行控制：用户希望能够控制他们的浏览器，并依赖那些遍布网站、展现连贯的数据。比如，他们不希望出乎意料地看到弹出的新窗体，他们希望能够使用“后退”按钮，返回他们刚才浏览过的网站。

4.2 使用整洁连贯的视觉设计

视觉设计告诉一个用户如何同网站进行交互。如果界面设计良好，用户将会花费较少的时间去学习如何使用一个网站，这样就有更多的时间去实际使用它。

不幸的是，我们在设计中经常把自己的创意放在重要的位置，以至于界面设计得太过于创新，导致用户在看第一眼时难以理解。具有创意无可厚非，但是当惯例让位于那些解决常规问题的新方法时，用户就不得不重新学习如何使用那些他们本已熟悉的东西。

当进行一个网站的视觉设计时，理解一些通用的惯例，会有助于创建一个看起来很棒而又不需耗费脑力去想如何使用的网站。

4.3 固定元素属于固定位置

当阅读报纸时，人们已经习惯报纸的顶部是报纸的名称，名称附近是日期，接下来是标题。图书在风格、素材和设计上可能会有些变化，但其中一些东西是可预知的，比如标题、作者和描述性摘要。

[1] 译者注：超级畅销书《点石成金:访客至上的网页设计秘笈》和《妙手回春：网站可用性测试及优化指南》作者。





报纸中，有一些人们凭直觉就能理解的惯例，比如标题和日期出现在顶部，重要的标题和作者的姓名出现在报道的附近。和报纸一样，网站也有自己的惯例，人们不需要解释便可理解。

类似地，当人们访问一个网站时也会有某种预期，这些预期如果没有被满足，就会比较困惑。因此，让用户预见到网站该如何操作，以及那些固定的元素会出现在什么地方，对网站设计来说是很重要的。

网站设计的一般惯例

- 链接到主页的网站标题（header）通常会在网站的左上方显示。不管用户在浏览什么网站，如果想返回主页，这都是他们想要的设计。
- 宣传语（taglines）和二级标题（secondary headlines）通常紧靠在网站标题的下面。宣传语会迅速地传达出网站的用途。
- 网站的搜索框（search box）通常会出现在网站的右上方或中部的上方。
- 不同类型的网站也许会有一些共同的习惯。比如，当访问一个提供 RSS 订阅的博客时，用户几乎毫无疑问地会先去页面的右上方查看订阅选项。

4.4 搜索是简单的

搜索并不是一直像现在这样简单。以前为了得到满意的结果，用户不得不设

置各种各样的搜索参数。亚马逊（Amazon）和 Google（谷歌）这样的标新立异者建立了一种新的搜索标准：输入一些内容，点击按钮，然后让网站对你想要的东西进行处理。从此之后，搜索不再是从一个在网站间切换的艰难过程。更多时候，可以通过一个输入框和一个搜索按钮完成。

很多时候，偏离这个基本的设计也是合理的，但是至少在没有为用户提供额外信息的情况下，搜索功能也应该能发挥作用。比如，如果通过添加一个下拉菜单允许用户限制检索条件可以带来好处，那就通过各种方式添加一个下拉菜单，但是在没有使用下拉菜单的情况下，搜索功能也应该可以使用。



人们每天在 Google 上数百万次地使用这个搜索模式。确切地说，用户不需要思考就知道该如何操作这个界面。

这就是搜索框的设计变得很重要的原因。搜索框必须清晰可见、容易辨识，同时使用方便。有人认为搜索框不需要设计，毕竟，它就两个基本元素：一个输入框和一个搜索按钮。蹩脚的设计会有多大危害呢？好吧，一些事情会出问题。比如，输入框显示的文本不易阅读，或者输入框太短或太长。一些设计者喜欢采用最简方案，根本不提供“搜索”按钮，用户必须使用“回车”键。这可不是最好的可用性解决方案。

搜索框必须是一个框。你的访客不会阅读页面，他们只会浏览。针对搜索功能最常见的设计就是一个框，一个包含输入框而显得相对宽大的框。用户倾向于在网页中通过浏览寻找这个图样，这种观念已经根深蒂固，所以应尽量避免使用其他的设计，比如带链接的文字或不带输入框的按钮。

同样，搜索框也要简单。根据对可用性的研究，在默认情况下不显示高级搜索选项会显得更加友好一些。对于高级搜索，一些普通用户在使用它的时候会感到困惑。一项研究表明，大多数用户不知道该如何使用高级搜索或布尔搜索的语法。[1]

有很多地方可以放置搜索框，但是只有几个是正确的。对用户来说，最常见

[1] Search: Visibale and Simple http://www.useit.com/alterbox/20010513.html。

的地方就是网站中每个页面的中部上方或右上方。当用户通过 F 形浏览网站的时候，很容易发现它们。然而，一些博客倾向于把搜索框放在（左侧或右侧）工具栏的下面。这也许不是一个好主意，但很可能是出于广告的考虑。

4.5　链接要像链接

永远不要让用户去猜测一个元素是否可以点击。一般来说，使用颜色和下划线的文本是一个链接。颜色和下划线要一起使用，这个组合是辨识文本中超链接最优的形式。

在创建链接时值得记住的一些事情

- 可点击的元素应该突出。颜色并不总是能表示链接，因为一个网站通常有一些并不是链接的有色元素。这就是为什么颜色和下划线的组合是一种稳妥的表示链接的标志。
- 不使用颜色和下划线的组合使非链接元素显得像个可以点击的按钮。这将使用户对于什么可以点击感到困惑。
- 在固定的区域内和页面间尽量使用相同风格的显示链接。不要使用不同的颜色表示链接，也不要对一些链接加下划线，而另一些不加。
- 链接应该有不同的状态。当为链接设定样式时，注意鼠标悬停（hover）、被激活（active）和被访问过（visited）这三个状态。这样，用户操作时，便会很明了。

4.6　仅在必要的时候使用下拉菜单

下拉菜单并不是没有用处，他们在网上绝对有自己的一席之地，但它们也经常会被误用。一般来说，下拉菜单用来让用户从备选属性列表中选出一个属性。

因为下拉菜单通常占用很少的空间，对一些设计者而言，它们成了最佳的导航方式。就在这种倾向大行其道的时候，下拉菜单的缺点也暴露出来。

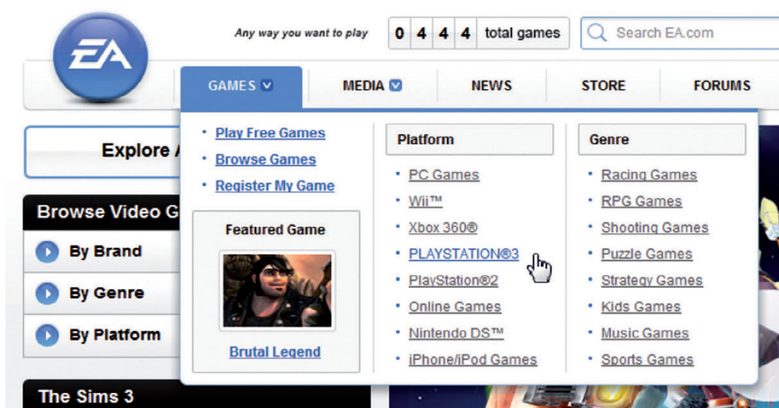
- 下拉菜单中的链接可见性较低，用户将只能看到一个选项，除非菜单被全部展开。

- 当下拉菜单包含太多选项时，用户不得不通过滚动来查看选项。在这种情况下，传统的列表会更好一些。
- 用户在凭直觉浏览时，下拉菜单不是最先被察觉到的东西。在选用它们之前与普通的导航菜单进行测试和比较，看它们的效率如何，不失为明智之举。

然而，下拉菜单并不一定要很小。根据 Jacob Nielsen 对下拉菜单的研究，所谓的“巨型”导航下拉菜单效果也不错。[1]很显然，大的二维下拉面板通过对导航选项进行分组，把用户从不得不进行的滚动中解脱了出来，通过有效地使用版式、图标和工具提示，它可以精确地呈现用户选项。

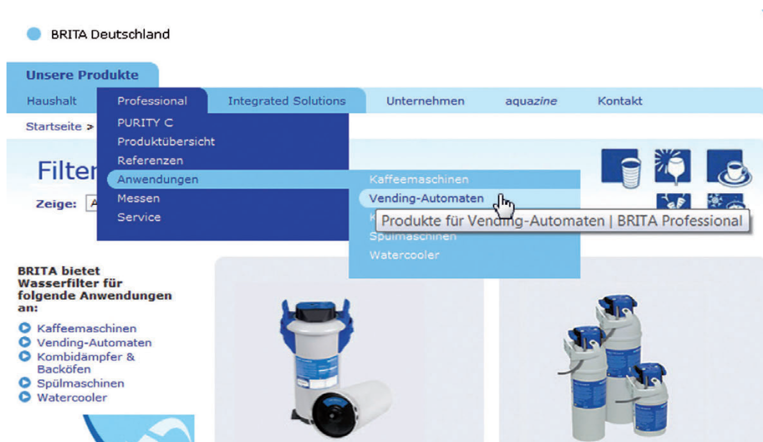
下拉菜单能将内容组织成整洁的小元素，如果没有做好，它将和糟糕的布局一样。当多于两个级别时应避免采用下拉菜单。如果子菜单通过悬停展开，只要用户的鼠标指针离开它们，就会失去焦点。如果子菜单通过点击展开，将需要太多的点击操作，使菜单难于使用。

下面展示的网站（Brita.de）犯了这样的错误。菜单太难使用，因为即使鼠标指针轻微失去焦点，你就得从头开始。注意工具提示，它挡住了导航信息。实际上，为了使子菜单可见而且可读，将下拉导航中的工具提示全部移除是个好主意。



一个漂亮、整洁、合理组织的巨型下拉菜单，来自 EA.com。注意紧挨着“Games”和“Media”列表项目的可视箭头（visual indicators），它们清楚地告诉用户在下拉菜单中将提供进一步的导航选项。

[1] http://www.useit.com/alterbox/mega-dropdown-menus.html。



Brita.de 的下拉菜单很难使用，它包含了多个层级的导航，必须进行水平滚动，同时工具提示挡住了导航选项。界面不够友好。

避免使用多个下拉级别，这对于你正试图创建的导航类型来说，也许是个太过严格的限制。然而，有一个解决方案也许可以增进悬停功能和多级导航的可用性。一旦用户把鼠标指针移开菜单，下拉列表连同大部分菜单会立刻消失。这个解决方案是延缓它们的消失过程。或者增加一个单击功能，要求用户点击下拉菜单外部的区域来关闭它，这和遮罩层（Lightbox）的工作方式类似。

仔细看一下保时捷（Porsche）网站的菜单。它有多个层级，但是它的菜单有一个广阔的焦点区（focus range），所以你必须把鼠标移开一定的距离才能关闭它。这是一个很好的设计方案。



保时捷网站有一个大的巨型下拉菜单。因为菜单有一个广阔的焦点区，使导航变得友好易用。

4.7 使注册表单尽量简洁

设计高效的网页表单并不容易，很简单，没人喜欢填表单，无论是在线上还是在线下。除非你的产品或服务具有革命性的理念，使用户看第一眼就被打动，否则把任何一种注册表单堆砌到你的网站上都是无济于事的。做出合理的设计决策，使用户完成表单变得轻松、容易、舒服是很重要的。

根据我们“注册表单设计调查”（Sign-up Form Design Survey）[1]的结果，注册链接最常用的标题是“注册”（Sign up），并且会放置在页面的右上角。注册表单通常有一个简单的布局，以防分散用户的注意力。它们通常会占一个页面，并通过解释注册将会带来什么么好处来吸引用户注册。

输入框通过粗体突出（62%），输入标签（当它们放在输入框左侧时）通常右对齐。表单的设计通常要尽可能地简单：设计师倾向于使用少量强制性的输入框（3~5 个）和少量可选择的输入框（2 个或更少）。垂直排列文本框要比水平排列好一些。还没有一种标签排列方式（比如放在右侧、左侧或输入框的上面）成为主流。

注册表单通常不包含分散注意力的内容，这样有助于使用户专心完成注册。

Password
(Letters and numbers only please. No spaces.)
57%

(Letters and numbers only please. No spaces.)
Password
13%

Password
4%

Password
26%

提示通常出现在输入框下面（57%），或者放在右边（26%）。

Label / Longer label / Even longer label / One more label / Value 1 / Value 2 / Select Value / Primary Action
62%　20%　18%　0%

输入框的标题用粗体突出（62%），输入标签通常右对齐。

[1] http://www.smashingmagazine.com/2008/07/04/web-form-design-patterns-sign-up-forms/。

它们通常不包含悬停、激活和焦点等状态。提示通常放在输入框的下边(57%)或者放在输入框的右边（26%)。

电子邮件地址通常不需要确认（82%)，但是密码通常需要确认（72%)。若不使用表格形式，“取消”按钮几乎没有用，“提交”按钮左对齐（56% ）或居中（26%)，然而 Captchas 使用了完全不同的表单。最后，感谢信息通常用来激发用户进一步对网站的服务进行探索。

4.8 考虑添加“返回页首”链接

不幸的是，让用户返回到页面的顶部——这个友好的服务，很少有网站提供。很多设计师的作品中都没有包含它，这是因为“返回页首”链接很少被使用。比如，对于那些页面或文章短小的网站，它显得毫无必要。对于内容可以显示在屏幕之内页面，点击这样的链接，页面不会有任何变化，这有时候会招人厌烦。显示器分辨率的多样化使“返回页首”链接有时变得没有用或没有必要。

然而，对于那些页面很长的网站，提供这个极好的功能既可以节省时间，又可以省去鼠标的垂直滚动操作。

“返回页首”链接的主要问题是它影响浏览器的导航功能并且会“污染”浏览器的历史记录。因为像其他链接一样，这个链接使用了锚，点击浏览器的“返回”按钮，它会把用户带回到本页面的底部，而不是前一个页面。另外，可用性专家声称，“返回页首”链接在概念上是模糊的，在网站之间很难连贯地完成它的功能，同时会影响到基于话语的用户代理的使用。

“返回页首”链接会在 FAQ、帮助和站点地图中使用，它有助于分割章节和段落，并为用户提供一个快速跳回到开始页的方式，而主导航就在开始页。

4.9 尽可能地避免使用 Splash 页面

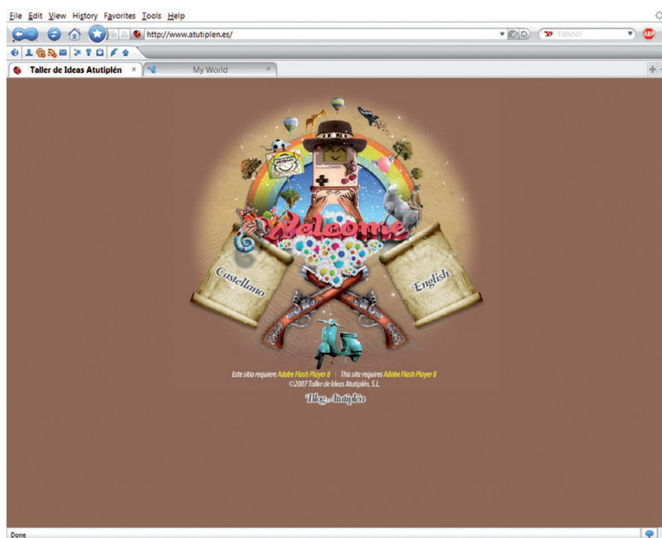
Splash 页面是网站的首页的一种，它除了让用户获得一些网站的信息和背景知识之外，不提供任何实质的内容。设计师在他们的作品展示网站中使用 Splash 页面是为了通过赏心悦目的效果吸引用户的注意力。使用它们，是为了把用户的注意力吸引到公司最新的产品上去。但是它们却增加了很长的加

载时间，除了“进入网站”之外，几乎没有任何导航选项。

凭借着设计师的创造力，Splash 页面可以提供很多吸引人的视觉元素。有一些使用了可以自动播放的交互式 flash 动画（对于打开了多个此类网站的用户而言，这简直是灾难）。

它们通常使用非常简单的结构，大多数只有一张图片、几行文字和链接。Splash 页面和网站的设计风格不会总是保持一致。

虽然大多数网站不使用它们，Splash 页面仍然很流行，因为有时它们是必需的，比如展示免责声明、重要信息（活动截止日期、重要升级等）、浏览网站的软件要求（Flash、Java、QuickTime 等）或浏览提示（通常用于 Flash 网站）。Splash 页面也可以用来让用户选择他们喜欢的浏览模式（标准或者全屏）、网速（高或低带宽）以及语言（虽然语言选择可以也应该在主布局中整合）。底线是：用户不喜欢 Splash 页面，“创造性”的 Splash 页面会惹人烦。



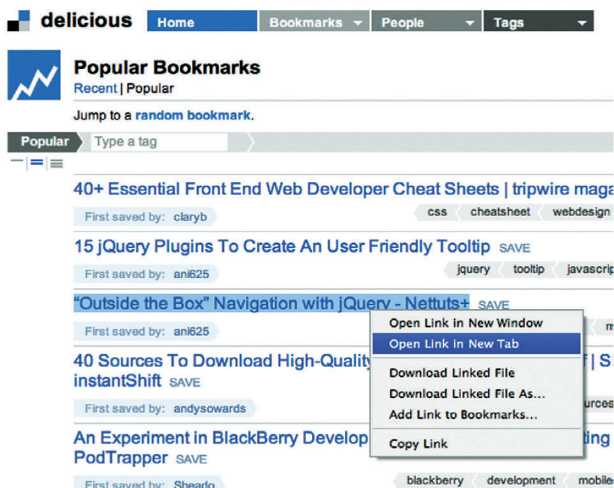
一个典型的 Splash 页面，包含一条欢迎信息和语言选择链接。

4.10　链接应该在同一窗口打开

最初，是否要在新的窗口打开链接似乎取决于个人网站和用户的喜好。访问

那些包含很多外部链接的网站时，用户更希望能够自动在新的窗口打开这些链接而不是必须手动完成这一过程。访问那些包含少量外部链接的网站时，用户希望能够选择哪些链接要在新的窗口中打开，以便稍后浏览。 然而，事情并不总是这样。

从可用性的角度考虑，强制在新的窗口中打开链接破坏了 UI 设计的一条基本原则：用户应该始终能对界面进行控制。用户需要界面具有连贯性，用户需要知道他们的交互活动不会被打断。任何偏离了这条原则的做法都会使网站变得更加设计导向（design-oriented）而不是用户导向（user-oriented）。正如 Shneiderman 所说的那样，有经验的用户有强烈的欲望希望知道他们正在控制系统，以及系统将会对他们的动作做出反应。



用户应该能够选择是在当前标签页、新标签页，还是在新窗口打开链接。

强制在新窗口打开链接的主要问题是，这个决定淹没了用户自己控制浏览器行为的决定。考虑一下用户不知道该在新的窗口还是在同一窗口打开链接的两种情况。

1. 用户希望在新的窗口打开链接，但是网站却在同一窗口打开链接。
2. 用户希望在同一窗口打开链接，网站却在新的窗口打开链接。

第一种情况，用户可以通过上下文菜单或快捷方式选择在新的窗口打开链接。

在这里，用户是动作的发出者，因为他或她决定如何展示链接所指的页面。网站的行为满足用户的预期，用户获得了良好的体验。

第二种情况，用户点击链接，仅仅发现网站跳转到了新的页面。在这种情况下，用户仅仅是动作的接受者，无法控制打开新页面的行为（最多只能当场关闭它）。网站的行为不符合用户的预期，用户获得的不是很完美的体验。

当一个网站的行为不可预期时，用户会被惹恼。如果在新的窗口打开链接会带来更多的便利，那么就让用户自己去完成这一动作。不要通过替用户来做决定的方式来侮辱用户的智力。不要强迫用户打开新的窗口，除非你有非常充分的理由。

4.11 你需要注意以下可用性检查点

1. 不要改变用户窗口的大小：这个观点和反对弹出窗口（pop-up）的观点是类似的。一些浏览器，比如 IE，会为以后的进程保存浏览器的尺寸。正如 Ben Bodien 所说的那样“认为自己比用户更清楚软件环境该如何配置是非常不合情理的”。
2. 不要使字体太小：字体如果太小，长篇的文章阅读起来会比较困难。也不要把链接、表单、搜索框和其他的元素弄得太小。另外，确保任何内容块都在 50~85 个字符之间，在文字的周围提供充足的空白。
3. 链接文本要清晰：链接文字应该用来描述它所指向的目的地。避免使用含糊的描述。
4. 去除死链，网上有太多死链了。不要把你的用户引入死路。
5. 校对：通读全部内容。即使你已经读过了，再读一遍。另外，也请别人来读一读。你总会找到一些可以改进的地方。看看你是否可以在保持特性不变的情况下压缩文本。把大的文本块分成小的段落。添加醒目的标题，使用列表以便用户扫读。校对时也不要忘记动态文本，比如弹出窗口的文字。通过检查，确保重要的文字在显示时不会成为孤立的一行，确保标点符号使用正确，尤其是撇号、引号、连字符和破折号。
6. 检查功能：仔细检查每一处。如果你有一个建立联系的表单，测试一下它的结果。让其他人去检测你的网站，不仅仅限于家庭成员和朋友，还

应包括目标用户。坐下来看看他们如何使用你的网站的，当其他人使用网站的方式和你的预期不同时会发现什么，这是个很有吸引力的问题。通常要检查的东西是建立联系的表单、搜索功能、购物框和注册区域。

7. 使表单尽可能简洁：使用合适的表单元素（选择按钮、复选框、输入区域、文本区域）来完成相应的任务，提供充足的空白（以及足够的边距）以便输入内容，移除不必要的区域，为每个输入区域提供描述性的标签（如果有必要添加帮助信息和提示信息）。同样，不要强迫用户使用严格的输入格式（比如，当用户输入的电话号码没有括号、连字符和空格时，系统也应该能够识别）。同样，提供有用的、信息充分的错误提示信息。

8. 优雅降级。你的网站在 JavaScript 不起作用时也要能正常工作。因为出于对安全因素的考虑，用户通常会关闭浏览器对 JavaScript 的支持功能，所以对此要有所准备。通过测试，确保你的表达在这种情况下仍然能完成服务器端的有效性检验，同时对你使用的 AJAX 效果进行测试。

9. 不要忘记保护性设计：最容易被忽略的保护性设计元素就是 404 页面。当用户请求一个不存在的页面时，就会显示 404 页面。通过提供一个有用的 404 页面把你的用户引导回来，这个页面可以导向主页，也可以向其他用户推荐他们可能感兴趣的页面。同样检查你需要验证的表单。通过提交异常信息进行测试（在需要填数字的区域填写大量字符和字母等信息），看一看是否会发生错误，以及是否为用户提供了足够的反馈信息以便修正它。

10. 考虑打印网页：如果用户想要打印网站上的网页，他们只希望打印主要内容而不是导航和额外的设计元素。这就是为什么要创建一个专门用来打印的模板。同时，某些 CSS 元素，比如悬浮文字区（floated text block），打印出来的效果并不总是很好。

11. 为获得最佳性能而努力：你将会为了获得最佳性能而配置你的网站。在网站发布后应该持续进行这一工作，但是你也可以在发布前通过简单的几个步骤进行操作。减少 HTTP 请求、尽可能使用 CSS sprite、优化网站图片、压缩 JavaScript 和 CSS 文件，这样你的页面会载入得更快，并且占用更少的服务器资源。

12. 要让用户很容易联系到你：也许你不想与用户联系，但是如果你的用户

想要和你联系，但是却找不到任何联系信息，他们将失去兴趣和对你的信任。这对于网站而言是个灾难，它会使网站错失良机。

4.12　传达你的网站的组织结构

在《用户体验元素》一书中，Jesse James Garret 提出了网站骨架的概念。简单地说，网站骨架有助于决定哪些表单对网站的功能而言是必要的。以下是这一理论的一些背景。Garret 假定网站用户体验领域可以通过两个不同的角度进行探索：一个是软件界面（比如在网站应用中），另一个是超文本系统（这将是内容管理系统的内容部分）。

虽然一个网站很少能完全通过它的内容进行描绘，但也很少能完全通过它的功能进行描绘。不管情况如何，我们都总是必须传递它的组织结构。



Girlgamer.com 使用了主题颜色来对网站不同部分进行区分。不同之处虽然相对细微，但是却能让各部分很好地展示自己。

设计清晰易用的界面属于界面设计的领域。如果一个网站主要是由内容驱动的，我们就不得不帮助用户获得这些内容，这被称为导航设计。在以上两个实例中，我们都需要帮助用户弄清楚我们网站上的哪些信息对他们是可用的，这就需要用到信息设计。再重申一下：界面设计帮助用户做事情；导航设计帮助用户找到想要的东西；信息设计帮助网站与用户进行交流。在为一个网站进行视觉设计时，这些都需要考虑。

4.13　使用标记来构建上下文

每一部分使用不同的颜色、页眉和按钮，可以帮助用户定位。这些被称为标记。网站标记的作用很像购物中心和游乐园中信息牌上的“我在这儿”标签。这就是为什么在新的窗口打开链接降低了用户体验：用户访问内容的连贯性遭到了破坏。用户了解自己在网站中的位置会有助于他们建立一种目的感，并使他们问自己“我能在这做什么？”。这就是信息设计的目的：使用户能够对他们可及的信息进行选择。

考虑一下像日间看护这样的行业，它的网站必须能够吸引不同的用户。网站的一部分可以使用非标准的版式、离奇的文字说明、卡通文字和明亮的颜色来吸引孩子；然而另一部分应该使用优雅的字体、直接的说明、抽象的图案和柔和的配色方案来吸引成年人。通过这种方式，网站吸引了广泛的用户，并通过标志把不同的信息定位到不同的部分。

4.14　建立良好的导航

除了内容外，导航可能是网站最重要的部分。幸运的是，我们能够利用一些惯例来把设计决策和导航连接起来。在多数情况下，用户仅仅试图得到“我能去哪儿？”这个问题的答案。但是这个答案并不总是像他们希望的那样简单。总体来说，用户在使用内容为主的网站时，希望任何页面都能提供 3 种类型的导航。

4.14.1　全局导航（Global Navigation）：在整个网站中我都能去哪儿?

全局导航通常以表格式导航的形式出现，还会包含一个便捷的指向网站主页

的链接。全局导航应该包含“受控制的”文字和短语（比如各栏目的名称、行业标准术语和外行使用的语言），并且突出网站的结构。因为全局导航如果放置得体的话，将会成为用户最先去的地方，它的设计至关重要。

4.14.2　本地导航（Local Navigation）：在这个部分我能去哪？

许多用户喜欢进一步浏览和他们正在浏览的内容相关的内容，本地导航正好能满足这一需求。如果网站组织得很好，当用户访问某一栏目时会知道他们将会发现什么，本地导航就是他们获得此类内容的入口。



因为本地导航时用户在网站停留的一个共同的原因，很多网站现在采用了一种混合导航的方式。例如，在 Amazon 完成了一次搜索后，用户可以通过使用本地导航，按类别进行浏览，及时并且恰如其分地满足了用户的需求。

4.14.3　上下文导航（Contextual Navigation）：在这个页面我能去哪儿？

在用户已经深入到你的网站，并发现了他们正在寻找的内容，他们接下来该去哪了？上下文导航会把用户留在同一页面，但是允许他们通过容易理解的内容区域四处活动。

上下文导航是一种未被充分使用、但十分具有价值的导航形式。

当创建上下文导航时，把菜单放在内容的顶部（通常是一个目录表或文章分类列表），同时在页面的各部分都包含“返回页首”链接。如果执行正确，上下文导航会把你的网站上的页面变成资源，鼓励用户经常回来寻找信息。想一想维基百科每个页面是多么有用。更重要的是，上下文导航提供了一个独特的（但经常被忽略）机会，它可以在构建网站品牌的同时，进行交叉销售、促销和使用户获得资讯。

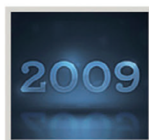
4.14.4 补充导航

在浏览你网站的一个页面后，许多用户离开了，访问了其他网站或关闭了浏览器。这时，一个好的主意是在内容的下面紧跟上补充导航。补充导航通常包含相关内容，以便用户能够停留并获悉你的网站还能提供什么。

面包屑（“面包屑路径”）是一种典型的二级导航模式，它用来定位用户在一个网站或一个网络应用中的位置。这个词来自于 Hansel 和 Gretel 的童话，童话中两个小孩通过扔面包屑形成的路径回到了自己的家里。跟童话里的情况类似，面包屑在真实的网站应用中为用户提供了一种返回到他们出发点的途径。

Now What?

You've reached the end of this post. Seeing you made it this far means you might be interested in these related articles and resources:



9 Design Blogs to Follow in 2009



Tutorial9 Presents: The Gift of Knowledge



Our Favorite Tutorial Indexes, Directories and Feeds



The New Tutorial9

在读了 tutorial9.net 的一篇文章后，读者面前呈现的是一个网站相关文章的列表，这就是上下文导航的实例。

在那些包含很多按级别排列内容的网站，你经常会发现面包屑。你也会发现他们在网站应用中不仅仅只有一步，他们的作用像一个进度条。最简单的形式是由水平排列的，并由“大于号”（>）分割的文字链接构成，这个符号显示了当前页面的级别。

因此，如果你的网站很大并且需要按级别排列页面，使用面包屑导航是一个很好的主意。电子商务网站是一个极好的例子：大量不同的商品按照逻辑进行分类。对于没有任何分级和分组、只有一级的网站不应该使用面包屑导航。用来决定一个网站是否可以从面包屑导航中获益的一个方法是建立一个站点地图或网站导航结构图，并且分析面包屑导航能否提高用户在不同类别之间导航的能力。

面包屑导航应该被看做一种额外的导航特例，不应该取代基本的导航菜单。它是一种便利，一种二级导航方案，它可以提醒用户他们在哪里，并提供一种不同的浏览网站的方式。



苹果网店在主导航的下面使用了面包屑 。这是补充导航的一个例子。

通过为一个多层级的网站的全部页面提供面包屑路径，用户可以通过很少的点击就可以返回到更高级别的分类，这比使用浏览器的“后退”按钮和基本的导航要方便得多。另外，因为它们通常水平放置并且样式简单，面包屑路径不会占据太多页面空间。

面包屑路径是一种吸引第一次来访的用户通过登录页（landing page）细读网站的绝佳方式。如果用户通过 Google 引擎搜索到了某个页面，看到面包屑路径之后，它们可能会点击更高一级的页面去看相关的主题。作为回报，降低了网站的整体跳出率。

4.15　重要的易用性法则和原则

4.15.1　7 ± 2 原则

因为人脑处理信息的能力是有限的，所以它通过把信息分成块和单元来处理复杂问题。根据 George A. Miller 的研究，人们短期记忆每次能处理 5~9 件事情，这经常被作为把导航菜单的选项限制在 7 个以内的依据。但是关于“7 ± 2[1]”的争论很激烈，以至于我们并不完全清楚能否把它应用到网站上。

[1] http://www.ddj.com/184412300。

4.15.2 两秒钟法则

这是一个宽泛的理论，即用户在使用某类系统时的等待反应（比如：功能切换和功能载入）的时间不应该超过两秒。两秒的推论稍显武断，但是这却是一个合理的数量级。一个更可靠的原则是，用户等待的时间越短，用户体验越好。

4.15.3 3 次点击法则

根据这条原则，用户在 3 次点击之内如果还没有找到他们想要的信息或了解网站的特色，他们就会离开该网站。这条原则突出了清晰的导航、符合逻辑的结构和易于理解的网站层级的重要性。在大多数情况下，这与点击的次数是无关的，但是用户始终知道自己现在在哪儿、从哪里来和他们能去哪儿是很重要的。如果用户觉得他们对系统如何工作有一个全面的理解，即使点击 10 次也没问题。

4.15.4 帕累托法则

帕累托法则（也被称为“少数关键定律”或“因素稀疏定律”）表明，80% 的结果是由 20% 的诱因产生的。这是商业中一条基本的经验法则（80% 的销量来自 20% 的客户），但是也同样适用于设计领域和可用性领域。比如，对那些制造出 80% 利润的 20% 的客户的关注度提高到最大，就会对你的收益产生巨大的改善。

4.15.5 界面设计的 8 个黄金法则[1]

Ben Shneiderman 通过对用户界面设计的研究，提出了一系列适用于大多数交互系统的准则。这些准则广泛应用于用户界面设计，严格意义上说，对网站设计也适用：

1. 保持连贯性；
2. 允许“老鸟”使用快捷方式；
3. 提供信息反馈；
4. 以对话框结束操作；
5. 提供简单的错误处理；
6. 允许轻松返回上一步；

[1] 译者注：更多信息请参考http://www.cs.utexas.edu/users/almstrum/cs370/elvisino/rules.html。

7. 满足用户的控制需求；
8. 降低短期记忆负荷。

4.15.6　费茨定律

由保罗费茨发表于 1954 年，费茨定律是一个模拟人类活动的模型，以目标距离和目标规模大小为函数，预测了迅速移动到目标区所需的时间。该定律通常应用于访客必须从 A 点移动到 B 点时的鼠标移动。例如，这条定律对于如何放置内容区域、以更实用的方式、最大化内容可及性和提高内容点击率，是非常重要的。

4.15.7　倒金字塔法则

倒金字塔是把摘要放在文章前面的一种写作方法。这种方式使用了新闻业中著名的“瀑布效应”——新闻作者试图让他们的读者即刻知道他们报道的主题。文章以结论开头，接着是关键词，最后是那些次要的细节，比如背景资料。由于网络用户需要即时的满足，这种倒金字塔风格[1]对于提高用户体验是非常重要的。

4.16　给用户一个 5 万英尺的视角来看你的程序

当第一次到访你的网站时，一些用户可能会感到无所适从。他们要对你的网站传达的东西有一个整体的了解并不容易，除非有效地展示了你的信息。我们应该仔细考虑一些组织内容的方法。在开始做这个之前，让我们考虑一下与之相关的行业。

专业的建筑师创建能够完成某种特定功用的结构。如果一个建筑师的任务是建一座礼堂，他可能需要从研究声学开始。演播大厅要怎样设计才能达到最佳的听觉效果？给预期的功能一种确定的形式，这是雇佣建筑师的目的。

现在考虑一下图书管理员的任务：组织海量的信息。简单地说，一个图书馆就是一个储存信息的地方。但是设想在一个无序的图书馆中寻找某本书的情况。

现在让我们回到网站。组织任何有很多内容或很强功能性的网站，这些方面的知识都必不可少，但是在网站中，我们把它称为信息架构（Information

[1] http://www.useit.com/alertbox/9606.html。

Architecture，简称 IA）。信息架构就是对网站的内容进行组织，以便达到最佳的信息传递效果。让我们看看一些可以向用户传达网站理念的方法。

4.17 心理模式和象征物

用户访问你的网站也是来完成某项任务的。他们会欣赏网站视觉设计和美妙的图像，但是在第一次到访后，他们只希望按日常习惯使用你的网站。

因此，考虑一下用象征物来展现你的服务。象征物是一种介绍你的产品或服务的捷径，几乎不需要解释，好的象征物已经把你的产品或服务和用户头脑中的信息（或心理模型）结合起来。

4.18 组织网站的内容

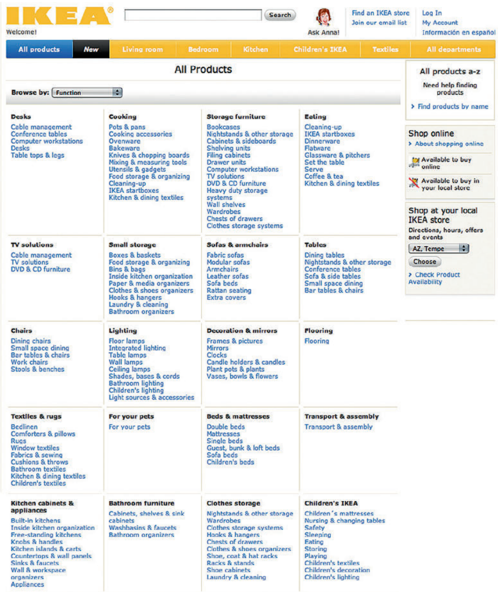
人们在头脑中会有日常生活中事物如何发挥功能的“路径图”。比如在某个早上，一个人知道他必须听到闹钟后起床、吃早餐、读报、洗澡、穿衣，然后开车去上班。所有的这些过程都是每天早晨要做的事情的一部分，人们每天都会重复这一过程。



亚马逊网站允许你把要购买的东西放入“购物车”（shopping cart），但实际上它仅仅是公司数据库中的项目列表。亚马逊为什么这样做？因为对于复杂的技术细节而言，购物车是一个很方便的象征物，它可以在你浏览网站的时候保存你购物的列表。当然如果你浏览的是 Amazon.co.uk，它会被写成“购物筐”（shopping basket）。

如果你的网站包含大量的内容，则需要寻找一个合理的方式组织它。进行分类是一个好的起点，也就是把内容分成语义相关联的组。这个理想解决方案只需要极少的开支，但却对用户非常有用。按读者的预期设计一个合理的分类方案，建立一套基于文字和短语的分类法。

另一种组织内容（以及优化网站搜索结果）的方式是使用元数据，也就是关于数据的数据。“元数据”听起来有点难懂，但是实际上你已经与它打了很长时间的交道。比如，一篇博客文章的元数据包括作者、出版日期、浏览次数和评论次数等。这类数据允许用户对网站的内容进行搜索和排序，这将为他们提供可以操作的内容列表，进一步逼近他们的目标。



宜家的网站很好地使用了导航。在网站顶部，宜家的狂热粉丝可以通过点击“New”快速找到最新和最热的家具。如果你正在寻找某类产品，宜家会为你提供一个结构清晰的站点地图。

4.19　提供站点地图

如果你的网站包含很多内容，站点地图是一种很好的导航方式。站点地图就

是一个页面或栏目，它以直观的方式展示网站上可以访问的页面。考虑一下把这个结构展示在页脚中或提供指向它的链接。

4.20 使用帮助

提供“使用帮助”是一种最直接的向用户介绍网站的方式。不幸的是，这也是你的网站上最可能被忽略的部分。大部分用户会直接忽略那些对服务进行描述的大块文本。

正因如此，一些有助于向用户展示网站如何工作的最佳实现方式脱颖而出，这些实现方式不依赖于大块的文本。这里有一系列提供使用帮助的方法，它们使用很少的文本或根本不使用文本。

4.20.1 白板

白板提供了一种可视化的引导用户的方法。这个想法很简单：展示一张图片，它显示了在包含数据的情况下你的网站看上去是什么样子的。看了这个如何使用网站或是应用的例子，用户就会知道该怎样与之进行交互。

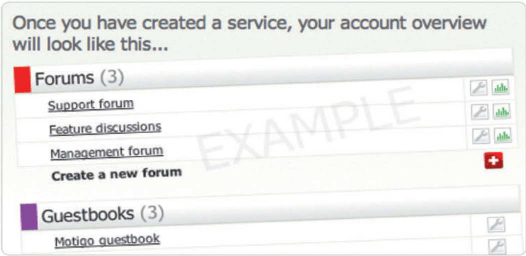
4.20.2 产品介绍和内嵌帮助

可以考虑把产品介绍融入到界面的各个部分。现在流行的多栏布局，使网站包含这类文档变得比以往任何时候都容易。如果用户对某个特殊的功能有疑问，他不用离开当前页面就可以获得答案。

4.21 建立信任

信任是你的网站在用户心中必须营造的最重要的事情之一。信任是一条基本原则，难以表述但是极为重要。很多研究将正面的用户体验等同于对网站本身的信任感。

建立信任：想一下生活中所有你信任的东西，并问你自己，“为什么你信任它们？”。考虑一下可信度、透明度和慷慨程度。尊重你的用户，显示出你对他们的信任和专注的感激之情。



Motigo 的用户在他们添加内容之后能够轻而易举地看到他们的操作界面将是什么样子。

4.22　不要让用户毁掉自己的用户体验

幸运的是，现代网站通常会遵循这条建议。虽然它主要应用于互联网应用程序中，但是它对那些内容驱动的网站同样适用。用户会测试你的应用程序的极限。如果用户可以删除内容，那么请为他们提供一个“恢复”功能。如果用户可以在输入信息时关闭浏览器，在没有询问之前不要抛弃他们的数据。如果用户在保存文档的过程中可以关闭浏览器，不要删除保存的数据。在这些案例中，要考虑到用户不会按照你预期的方式使用你的服务，因此尽量不要使结果让人无法接受。

不要要求太多而给予太少

在为设计最新、最好的用户交互努力的过程中，许多设计师忽略了作为有经验的设计师应该遵循的一条原则：不要破坏任何事情。一些设计师知道的比这个要多，但是现代网站设计却并不是没有遇到阻碍。所以让我们澄清一下：

任何作为用户浏览器一部分的功能，都不应该偏离用户使用网站的经验。

最基本的元素通常被最先破坏。不要屏蔽用户的“后退”按钮；不要限制用户为你的网站加书签的能力；不要在并非必要的情况下修改用户已浏览和未浏览的链接的颜色。很多刚入门的用户和中级水平的用户很大程度上要依赖这些特性来浏览网站。浏览器的这些功能对他们的体验十分重要。

5	TOP	Everything Is Illuminated	★★★★☆	Drama	Now	×
		Robin Hood: Men in Tights has been removed				Undo
7	TOP	Secretary	★★★★☆	Independent	Now	×
8	TOP	Sixteen Candles	★★★★☆	Romance	Now	×

当你在 Netflix 序列中的一项内容被删除后，网站提供了恢复操作的服务。

总体来说，一种更具渗透性的不信任感正在交互设计师及其用户之间酝酿。许多交互活动应该跨浏览器进行测试，在那些能被用户屏蔽的功能不起作用的情况下，交互活动仍能表现良好。不言而喻，一个好用网站的前提应该是一个可以使用的网站。

最后，要让用户知道你的网站是否支持他们正在使用的浏览器，不要让他们自己去猜。只有在你告诉他们之后，他们才会知道他们也可以应征加入浏览器的战争。

4.23 及早测试、经常测试

所谓的 TETO 原则应该被应用到每一个网站设计项目，因为可用性测试经常会为布局的问题和难题提供至关重要的参考。

测试不要太晚、测试不要太少，也不要为错误的理由进行测试。测试要有正当的理由，大多数设计决定都是局部的，理解这一点很重要。你不能准确无误地回答一个布局是否优于另一个布局，因为你需要从特定的视角进行分析（要考虑需求、股东和预算等因素）。



MailChimp 在它的主页展示了一个视频的范例，以便用户能够快速学会如何使用该网站。

一些需要记住的要点

- 根据 Steve Krug 的观点，测试一个用户一定要比一个用户都不测试好，在早期测试一个用户要比在后期测试 50 个用户好。根据 Boehm 第一定律，错误最常发生在需求阶段和设计阶段，越晚发现这些错误，付出的代价越高。
- 测试是一个反复的过程，这意味着你设计一些东西、测试它、修正它，然后继续测试它。你可能会发现一些在第一轮测试时被忽略的问题，因为用户的注意力已经被其他问题吸引。
- 可用性测试总是会产生有用的结果。它们既指出了功能上和交互性上的错误，也指出了主要的设计瑕疵，它们从以上这两个方面为你的设计提供了有益的见地。
- 根据 Weinberg 定律，一个开发者不适合测试自己写的代码。这也适用于设计师。在为一个网站工作了数周之后，你已经再也无法凭借肉眼识别任何问题。你知道一个网站是如何构建的，因此你非常清楚它是怎样工作的，而测试者和用户并不了解。

底线：如果你想做一个非常棒的网站，你必须进行测试

4.24 总结

当我们开始着手改善用户体验时，我们面对的最大挑战是该从哪里入手！它可能会让人有种挫败感，但是却是一个好的开始。这里有很多你可以考虑的方式，来使你的网站显得更加有吸引力。

付出努力，最终你将会得到一个用户和你都喜欢的产品。我们希望在这里讨论的大多数观点可以应用到你的项目中去。

- 虽然创意个好东西，但是不要让创意盖过了设计过程本身。不要把界面设计得太过新颖，以至于一般的网络用户难以明白如何使用它。
- 创建可预知的网站，设计用户期望的东西。不用试图使用未来网站的特色，除非它对完成旧的任务具有明显的优势或更加直观。
- 用户会快速浏览网站，不会停下来看那些设计者花了很多时间精心雕琢的细节。如果你能够使他们更加容易地从一处跳转到另一处，或帮助他们找到自己需要的东西，他们会更加欣赏你的工作。记住，很多用户仅仅希望把使用网站当做日常事物的一部分。
- 在投入视觉设计之前，花一些时间规划你的用户将如何使用网站（界面设计），他们会如何在网站上进行导航（导航设计），以及网站将会如何与他们进行交互（信息设计）。
- 不时用标识告诉用户他们在网站中的位置。知道自己方位的用户更容易建立目标感。
- 沿用导航惯例。如果用户不清楚他们能够去哪里，在进行导航的时候将会遇到困难。在设计导航的时候，不要停在第一个层级：为每个网页都设计全局导航、本地导航和上下文导航。通过让内容按语义关联进行分组，来帮助用户找到他们正在寻找的内容。
- 总是预想用户不会按你的预期使用你的网站。设想一下他们会如何误用你的网站，不要让误用造成难以接受的后果（比如，在没有提供“恢复”选项的情况下删除一个文档）。

网站设计、可用性和用户体验中超炫色彩终极使用指南

第5章

Dmitry Fadeyev

色彩在网站设计中如何吸引用户注意力，为决策提供信息，引发行动，组织信息，突出元素，平衡结构和丰富用户体验？本章将探索这些问题并列举一些好的网站色彩使用实例。

原始需求

研究发现，因为早期区分食物、寻找配偶、躲避敌害和危险环境的需要，原始人的色彩视觉进化到了高级敏感的阶段[1]。如果没有色彩视觉，区分毒性致命的绿色马樱丹[2](lantana camara ）浆果和不太致命的红色马樱丹浆果将会变得非常困难。

我们现在收集、体验、分析、筛选信息的速度远远高于我们的祖先，尤其是借助网络和其他媒体。设计中拙劣的色彩方案不会伤害任何人，但是它们却可以产生很糟糕的用户体验。因此，理解色彩如何塑造并改善访客体验和与设计的互动是很重要的。

打好基础

你可能想一下就投入到令人望而生畏的色彩灵感和色彩信息的世界，但是首先讨论一些色彩基础知识、色彩学理论和描述色彩的术语非常重要。一旦我们打好了这些基础，我们随处都可以对色彩进行探索并寻找灵感。

色彩入门

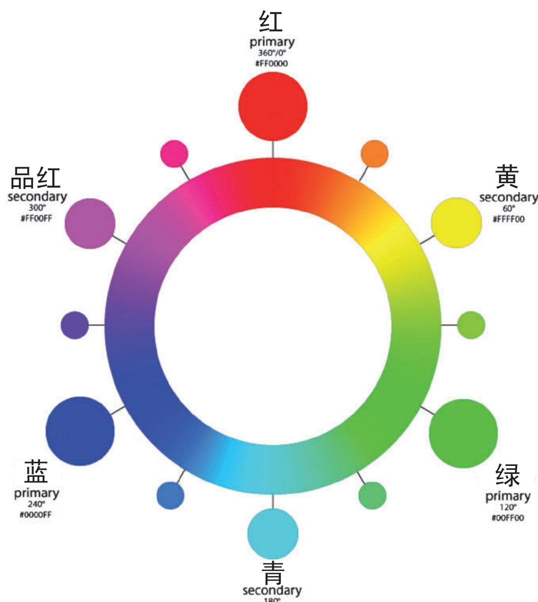
我们的眼睛有 3 种不同颜色的色彩接收器，分别对红、绿、蓝 3 种颜色做出反应。比如，蓝色对相应的接收器刺激要大于其他的接收器，结果是使它向大脑传送不同的信息。通过这种方式，不同色彩混合后会产生数以百万计的颜色，被称为色度（ hue ），影响不同的接收器。这些色调可以通过一个圆（ 通

[1] Regan, B. C., Julliot, C., Simmen, B., Vienot, F., Charles-Dominique, P. & Mollon, J. D. 2001 Fruits, foliage and the evolution of primate colour vision. Phil.Trans. R. Soc. Lond. B. 36, 229-283。

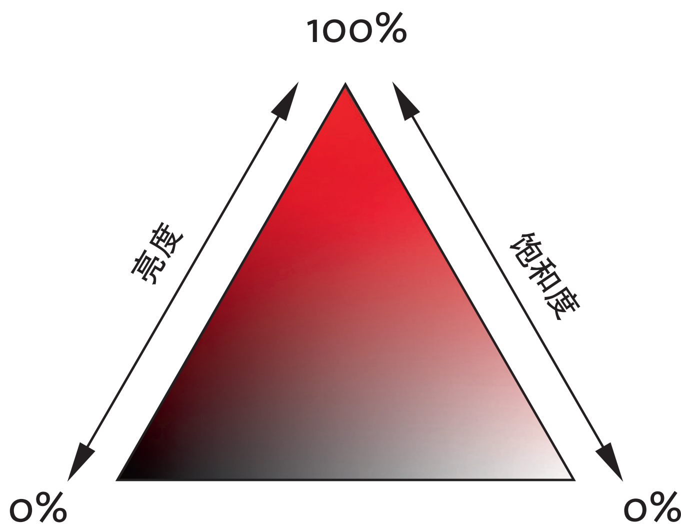
[2] 译者注：更多关于马樱丹的信息请参考：http://commons.wikimedia.org/wiki/Lantana_camara?uselang=zh-cn。

常被称为色盘）以度数的形式展示。每 60° 会产生一个不同的基色。这些基色中的 3 个被称为第一原色（红、绿、蓝），另外 3 个被称为第二原色（黄、青、品红），它们之间的 6 个基色被称为第三原色。

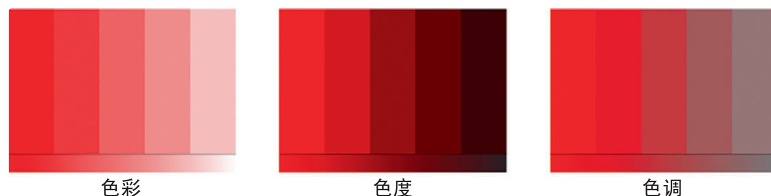
一个通用的计算模型被称为 HSB。它使用色度和其他两个相关的术语——饱和度（saturation）和亮度（brightness）——来定义和描述色彩。饱和度（0~100%）是指色彩的鲜明程度与它最大可能的鲜明程度的比值。亮度是指色彩的明亮程度（0~100%）。纯色具有最高的亮度和饱和度，使它们看起来更加鲜明。暗色亮度较低，使它们看起来暗一些。颜色越不饱和，看起来越淡，饱和度为 0 时完全没有颜色。正如你从电脑截屏上看到的那样，很多颜色可以通过一个色调值和不同的白、黑、灰色度（shade）合成。



在数字化设计中第一原色是红、绿和蓝。第二原色是青、品红和黄，在色盘上，两类原色之间是 6 个不同的色度。色度值用度来描述（°），用十六进制表示的前面会标上井号（#）。



通过调整色度的亮度和饱和度，大概每个色度可以得到 46000 种不同的颜色。



亮度和饱和度是比较复杂的概念，容易产生误解，这使它们在讨论颜色时不太常用。

也许讨论颜色时使用色彩（tint）、色度（shade）和色调（tone）这 3 个术语是有好处的。当一个设计师说“我喜欢这个 Logo 的颜色，但是它的色度应该更深一些”时，你会清楚地知道他是什么意思。

把色调、饱和度和亮度结合起来，我们就可以创建任何一种人眼可见的颜色。[1]很多计算机显示器通过红、绿和蓝三原色的混合呈现色彩。历史上，艺术家曾经使用红、黄和蓝作为原色。印刷工业使用青、品红、黄和黑来产生不同的颜色。计算机使用红、绿、蓝三原色，因为显示器发出这 3 种不同的光。通过对每种颜色在 0~255 之间的赋值，计算机

[1] Backhaus, W., Reinhold, K. & Werner, J. S. 1998 Color Vision, Walter de Gruyter & Co., New York, p. 321.

显示器能够生成16777216种不同的颜色。3种原色以最大值（R:255, G:255, B:255）混合时，将产生纯白色光，而3种原色的值都是0时，不产生任何光，也就是产生黑色。此类颜色混合通常使用的是RGB色彩模型。

浏览器可以识别十六进制以#号开头的网页色彩值。一种颜色的红、黄、蓝三原色值，由数字（0~255之间）转化成了一组计算机可以理解的字符。在每一色阶结束时，0被替换成了“00”，而255被替换成了“FF”。第一原色转化成十六进制之后变成了#FF0000（红）、#00FF00（绿）、#0000FF（蓝）。很多图形应用程序可以在HSB值、RGB值和十六进制值之间自动转化，但是作为一个设计师，理解每一个体系将有助于你更加轻松地完成工作。

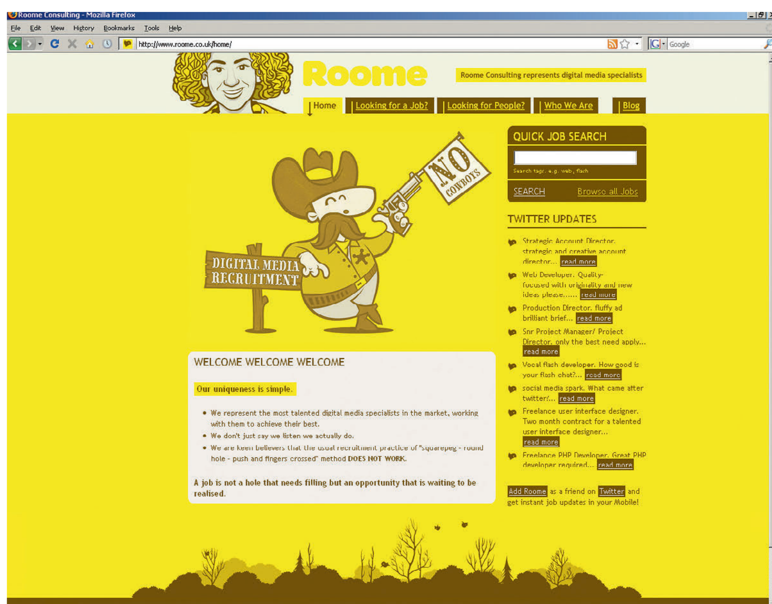
5.1　色彩学理论

色彩学理论研究的是纯色之间的关系，以及它们混合的效果。我们将研究一些使用最广泛的色彩关系或配色方案，同时给出它们在当前网络应用中的实例。设计一个配色方案通常是决定设计中将哪些颜色放在一块视觉效果最好的第一步。接下来的每一张截屏中，调色板显示了截屏中使用的主要颜色，纯色色调在色盘中被标记了出来，以便显示色彩关系。

单色：使用一种色调不同的饱和度和亮度



单色调色板

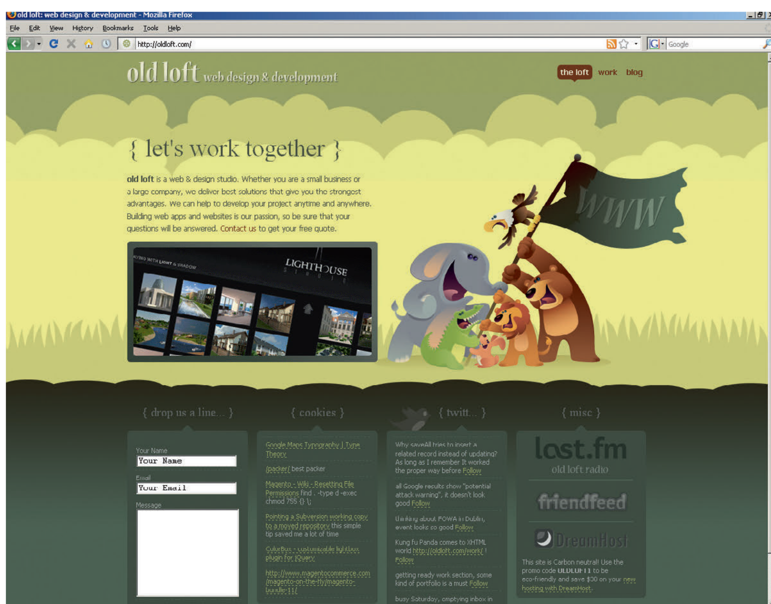


单色配色方案

近似或混合：使用色盘中邻近的色调



近似调色板



近似配色方案

互补：使用色盘中直接相反的色调

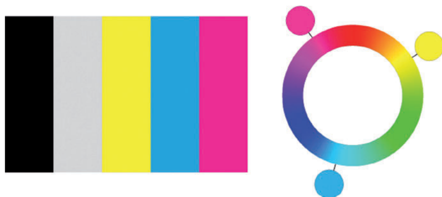


互补调色板

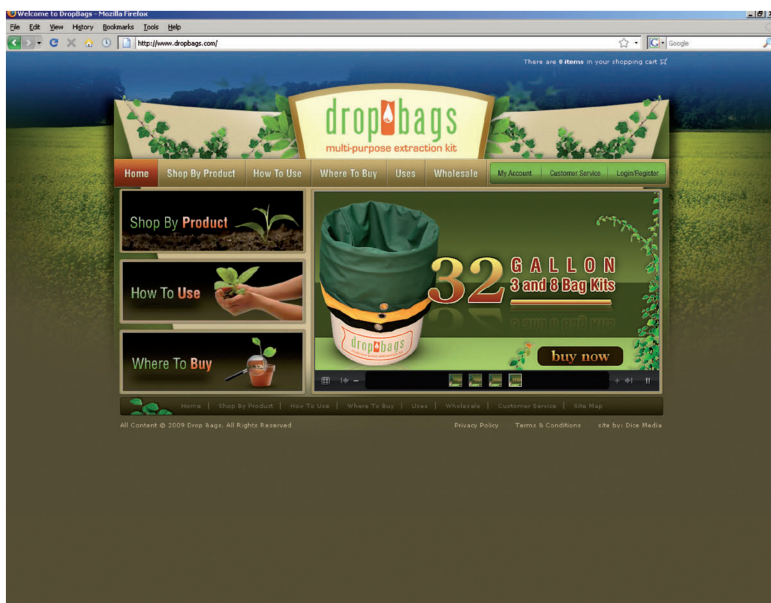


互补配色方案

分割互补：使用一个色调和两个与它的补色邻近的色调

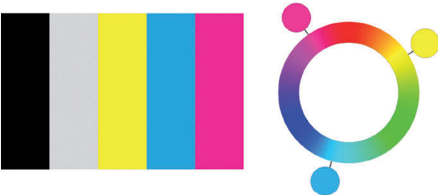


分割互补调色板



分割互补配色方案

三分色阶：使用色盘中彼此等距的三个色调



三分色阶调色板



三分色阶配色方案

双互补色或直角四阶色：使用两个色调和它们的补色



双互补色调色板

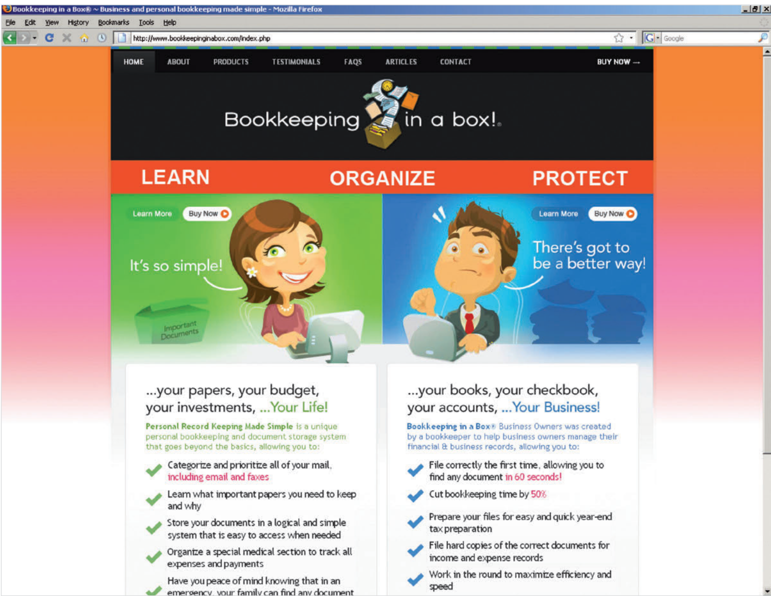


双互补色配色方案

四分色阶：使用色盘中彼此等距的四个色调



四分色阶调色板

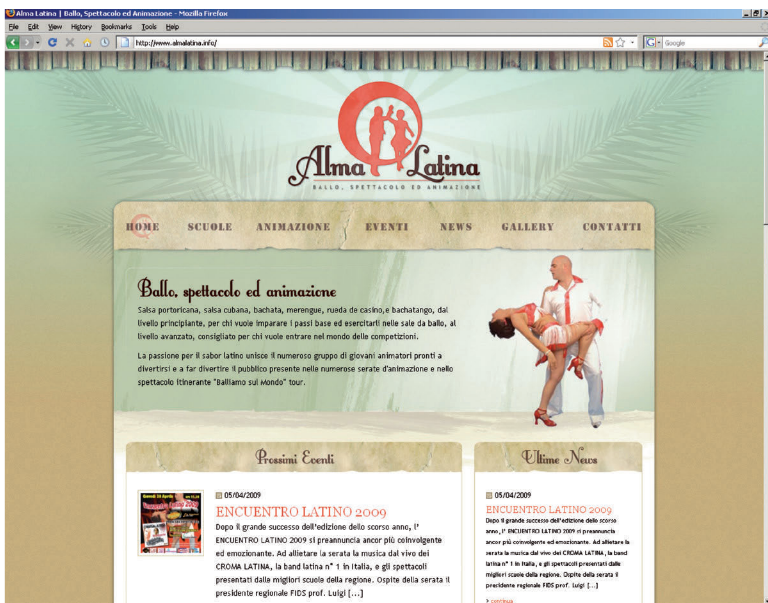


四分色阶配色方案

近似互补：使用两个互补色调和一个与其中一个颜色相近色调



近似互补调色板

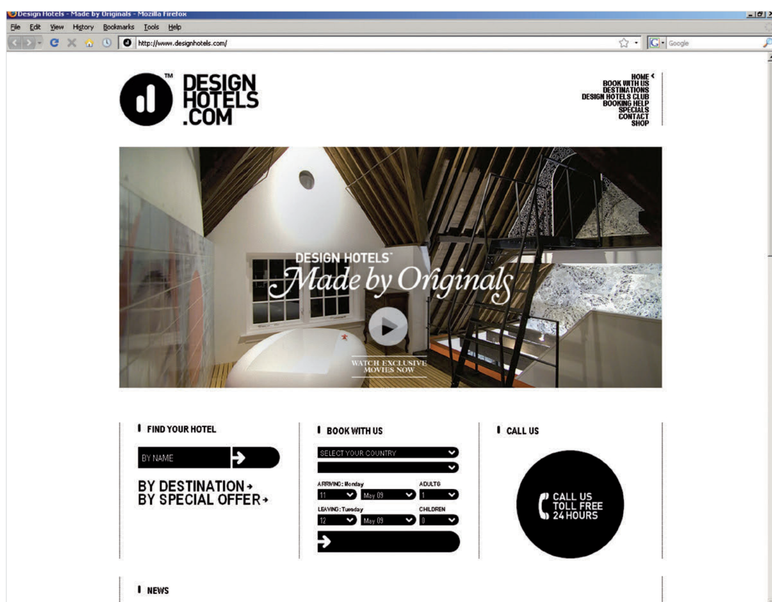


近似互补配色方案

中间色：使用没有色度的颜色



中性调色板



中间色配色方案

突出：在没有色调的颜色中突出一个高饱和度的色调

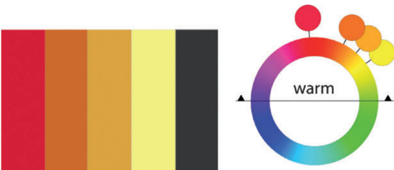


突出调色板



突出配色方案

暖色：使用色盘顶部的颜色，使用分布在 271° 和 90° 之间的色调

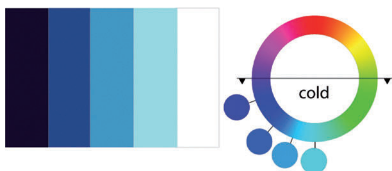


暖色调色板



暖色配色方案

冷色：使用色盘底部的颜色，使用分布在 91° 和 270° 之间的色调



冷色色盘



冷色配色方案

5.2　色彩和易用性

色彩不但可以使你的网站赏心悦目，还可以帮助读者使用网站。设想一下，如果你的网站还是白底黑字——就像早期报纸那样——情形会怎么样。在 1982 年，《今日美国》(USA Today) 率先推出了彩色报纸[1]。不但很多版面配有彩图，它还使用了色彩来区分板面，所以读者可以轻而易举地发现他们感兴趣的板面。对于新用户和回头客而言，这个理念在网站设计中都很重要。通过添加彩色的按钮，《今日美国》把这一理念应用到了它们的网站之中，这些按钮和纸版报纸的板面是相对应的。它为每个不同的版面选择了不同的颜色。然而不幸的是，“Home”、“News” 和 “Travel” 这几个版面使用了非常相近的蓝色色调，有时这可能会误导用户。



《今日美国》使用了色彩来帮助读者快速定位版面，它自 1982 年开行业之先河以来，一直坚持这样做。

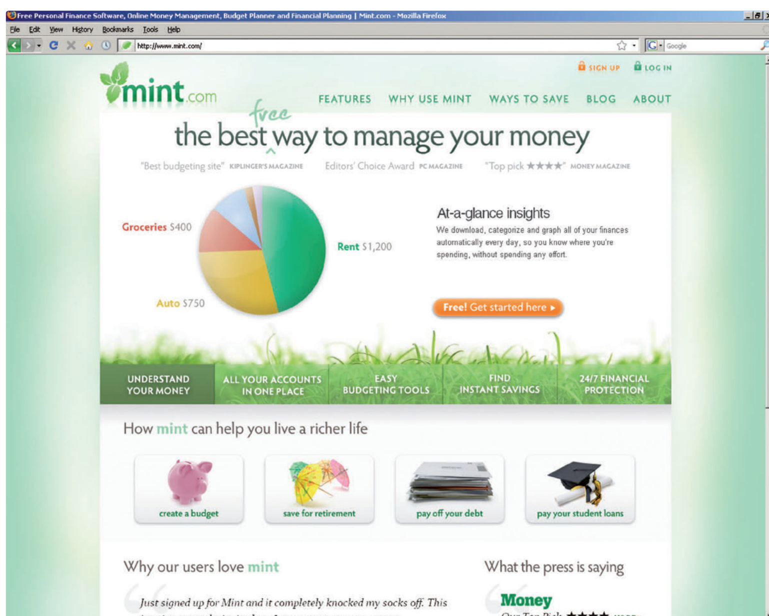
Mint.com 成功地使用了色彩来帮助潜在客户理解它的服务，并劝说他们采取行动。Mint 主页上最引人注目的色彩元素是使用了双互补色配色方案的个

[1] Garcia, M. R. Jun 18 2008 The color evolution of USA Today: a three-minute interview with Richard Curtis，http://garciamedia.com/blog/articles/the_color_evolution_of_usa_today_a_three_minute_interview_with_richard_curt。

人理财数据图表和写着“Free! Get started here”（免费！现在就使用）的亮橙色的按钮，它们在淡绿色的背景上显得尤为突出。



双互补色调色板



Mint.com 使用了一个双互补色配色方案来展示它的软件的用途。亮橙色的按钮在屏幕上显得很突出，会吸引住读者的眼球。

当表单出错时，这个网站还通过使用色彩来快速地给出反馈。它的创建者选用了红色，这个常用来提示危险的颜色，这个色调在淡绿色的背景下显得非常突出，很难被忽视。

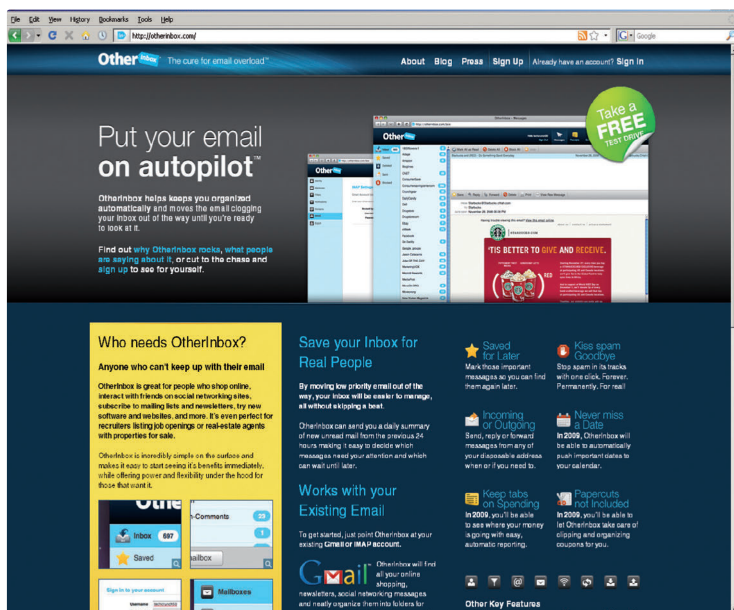
5.3　色彩与注意力集中

色彩能使用户注意力集中并哄着用户专注地使用网站。伴随着日常生活中时时存在的各种噪音和干扰，让用户专注地使用你设计的网站简直是一种非分之想。你应该用色彩把他们拉回到你的设计和内容中去，为此你可以使用色彩将他们的注意力吸引到你的网站最重要的部分。使用很少的色彩，UI Tuneu 快速地解释了它的服务，并指示你在哪里可以开始使用它的服务。

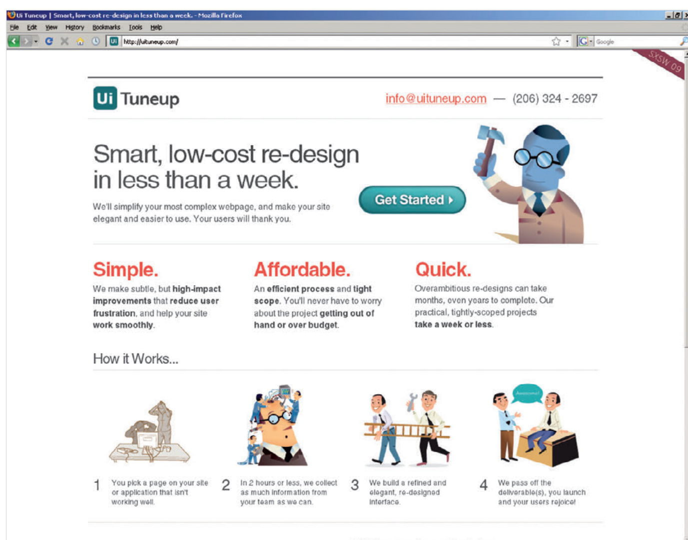




橘黄色和蓝色是近似的互补色。用绿色的标签脱颖而出，吸引注意力。



亮绿色的标签在相对柔和的暗色衬托下格外显眼。



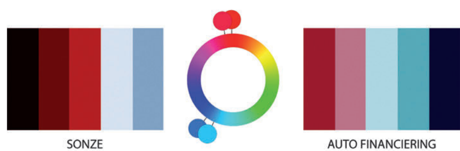
通过使用亮度较高的互补色，UI Tuneup 使用户很容易获悉网站的功能，以及如何开始使用这些功能。

即使你的网站使用了很多颜色，你也可以把用户的注意力吸引到某个元素上去，你可以通过给这个元素一个与众不同的颜色做到这一点。OtherInbox 使用绿色突出网站的焦点——“Test drive”按钮。

可能发生的最糟糕的事情是新用户无法理解网站的导航，并在感到迷惑后离开。色彩可以帮助他们从正确的地方开始，通过对比两个网站你就会发现这一点，它们基本使用了相同的互补色配色方案和相似的设计，但是却产生了完全不同的效果。

Sonze 使用了淡蓝色和深红色的按钮。但是由于使用的蓝色太淡了，如果用户发现按钮上有字的话，则需要很费力才能看清它写了些什么。同样，可以点击的项目使用了不太连贯的色彩，所以即使你发现了刚才说到的按钮，也不一定会找到其他可以点击的东西。

AutoFInanciering.nl 正好相反，它几乎通过深红色，突出了每一个可以点击的项目。因为背景较淡，用户很容易发现这些按钮。



这两个网站使用了相似的配色方案，但是却产生了相当不同的效果。



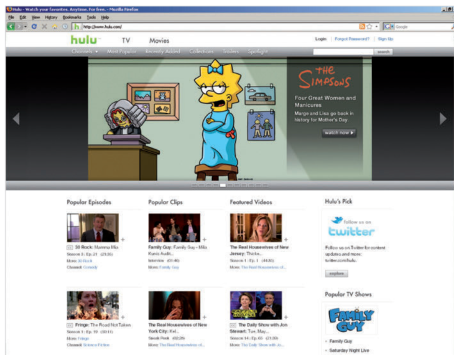
与 AutoFInanciering.nl 相比，Sonze 的按钮很难被发现，同时它可以点击的元素使用了不太连贯的色彩。



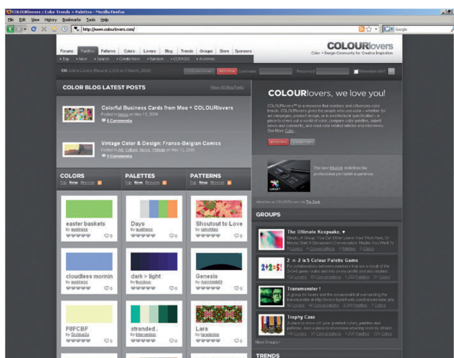
AutoFInanciering.nl 使用了连贯的色调，在不同元素间使用了不同的饱和度和亮度，这使用户可以很容易地使用网站。

对于一个包含大量内容的网站，内容的颜色持续变化的地方，使用中性的或着重突出的背景色有时是最好的选择。这样内容会在屏幕上凸现出来，不会和背景混为一体。很多摄影师在他们的作品展示网站中使用这种技术来突出图片。Hulu，一个流行的视频网站，采用了突出配色方案，我们自己的网站

COLOURlovers，用来创建并分享调色板的网站，也是用了这种配色方案。



Hulu 用绿色突出网站的 Logo，并使用了很少的颜色，以防止它们和内容冲突。



在 COLOURlovers 中，我们使用红色突出网站最重要的部分——社区部分。

5.4　色彩与体验

考虑网站所用色彩的文化内涵是很重要的，尤其是如果你希望网站能带来国际流量。比如绿色，它是一种常用的颜色，在西方世界常常跟环境意识相关；在中国“绿帽子”用来指妻子不忠；在伊斯兰世界，这种颜色是神圣的，在天主教中它也有自己的含义；在非洲一些国家中绿色代表富饶的物产。它也和金钱、嫉妒、生长、生病、没有经验、邪恶、肥沃、希望、青春和死亡相关。



调色板中的青色不适合任何一种刚才讨论过的原则，但是在图片中使用少许青色会产生非常美丽的效果。



这张取自 Paulo Brandao 作品集的落日照片[1]，向我们展示了大自然是如何使用色彩的。

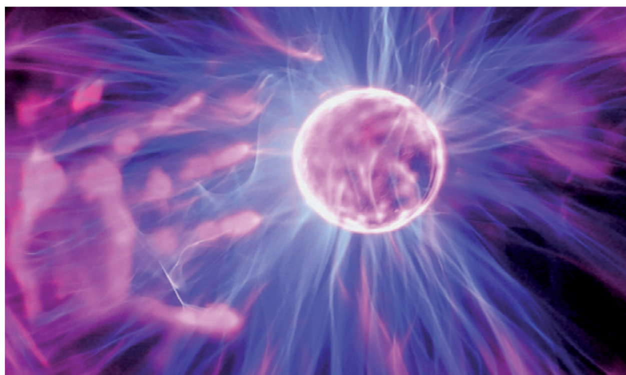
色彩也会影响人们的情绪和心理。在精神病院的墙上涂上一点浅绿色可以起到镇定的效果。吃绿色盘子里的食物会使人产生苦的味道；然而蓝色的盘子可能限制你的食欲；粉色的盘子会使你觉得食物更甜。[2]



这个取自离子球的调色板本身让人感到不安。但是通过艺术家之手把它放到背景之中，比如离子球，它可能变成一个美丽瑰奇的色系。

[1] http://www.flickr.com/photos/paulobrandao/2592581334/。

[2] Maeyama, M. Apr 2001 The Color of Yummy, Parenting Volume 15 Issue 3。



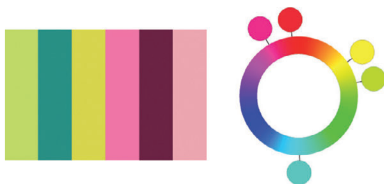
离子球是尼古拉·特斯拉（Nikola Tesla）的一项惊人发明。它的颜色仅仅通过激发球内气体的带电粒子产生。

色盘中所有的颜色都有自己的解读、民间传说和象征意义。所以，实际上并没有选择颜色的法则或秘方。在选色时，你还得依赖你自己的判断，什么颜色看起来好一些，什么颜色更适合你的项目。不要畏惧，坐下来看什么颜色最能打动你的心，而不是你的眼球。[1]

5.5　灵感

“优秀的艺术家复制；卓越的艺术家偷窃。”——帕布罗·毕加索

色彩在自然、科学和人类所创造的东西中无处不在。要想获得灵感，四处看看就行。很可能在你的手边就有一张图片、一个生锈的汽车挡泥板、一朵花和一只猫，它们可能包含了有趣的色彩组合。从你所发现的东西中提取色彩，你很可能会为这么容易就可以获得色彩灵感而惊讶。



这个来自糖果的调色板会使你的网站秀色可餐。

[1] http://www.flickr.com/photos/jurvetson/167197691/。



我们几乎在所有的东西中都可以发现美丽的调色板。一旦你选定一个调色板，按正确的比例展示色彩是接下来的挑战。在这张图片中，如果绿色代替粉色被重点突出，情况将会大不一样。[1]

我们希望这一章有助于你理解色彩学理论的核心概念和一些可以帮助你谈论色彩的术语，并给你一些去哪里为下一个具有创意的项目寻找灵感的主意。这些信息将帮助你的设计增加一些结构，并通过使用色彩影响或提高用户体验或产生共鸣。但是我们最希望你使用色彩时能够获得灵感。关于色彩的使用有很多规则，但是这些规则也等待着有人去突破，人们应该遵循内心的感觉和灵感。数字技术使人们在与他人分享激情时产生了一点距离，当你有这个广大的、多样化的市场时，这里有太广阔的空间充分展现每个人的创造力，使自己与众不同并创造一些美丽的东西。祝你们能够顺利地使用色彩并找到乐趣。



最初，这张关于一个围栏的照片看不出什么配色方案，但是当你开始抽取颜色时，一个不错的调色板出现了。虽然，它们可能看上去有点不协调，但是当把色调降下来后，这些纯色色调放在一起看上去很不错。

[1] http://www.flickr.com/photos/yomi955/783099682/。



侵蚀是大自然创造美丽调色板的另一种技巧。

网站优化　第6章

Rene Schmidt

响应缓慢甚至停止响应的网站非常恼人。如果你的网站惹恼了用户，他们就不会愿意购买商品或者为你的进步充分“陪练”，你会失去赚钱的机会。因此，你需要重点优化你的网站，以提供更好的用户体验。

Yahoo 的 Firefox 插件 YSlow 提供了如何提高网站响应的提示。本章不仅会介绍 YSlow 的这些技巧，同时也会深入讲解 MySQL 和 PHP 的优化技术。现在就开始吧！

6.1 你将会用到的工具

- 文本编辑器的选择。
- 配置好的 LAMP 服务器（Linux、Apache 或者 Lighttpd、Mysql、PHP5）。
- **FTP 账号**。FTP 是向服务器中上传文件的通用方法。因为这篇文章是面向具备一定经验的设计师和开发人员，我们假设您已经知道如何通过 FTP 访问你的文件。
- **SSH 账号**。为了修改远程 Linux 服务器的配置文件，你需要可以访问的 SSH 账号，可选的客户端有 Putty 或者 Jellyfish。我们假设您已经知道如何使用 SSH 客户端进行登录。你的网站需要开放 SSH 登录，这样才能获得 Web 服务器配置文件的修改权限。在很多共享主机上，往往没有这个权限。
- **MySQL 初级调试工具**：有下列情况之一的，你可以跳过本节。你没有访问 MySQL 配置文件的权限、你的服务器不是 Linux、你的数据库不是 MySQL。否则，请登录 http://launchpad.net/mysql-tuning-primer/ 下载最新的 MySQL 调试脚本。使用 FTP 或者 SSH 将脚本上传到服务器中，但是先不要运行。

- **安装 Smush**：这是一个 Firefox 插件，提供了大量图片优化的快捷方式，它是一个顺手的工具。登录 http://smush.it/，下载安装。Firefox 的 NoScript 插件也许会与它发生冲突，因此你需要禁用它（译者注：目前 Smush 被包含在了 YSlow 中，已经不需要再单独下载安装了）。
- **安装 Firebug 和 YSlow 插件**：Yahoo 提供了一个 Firebug 插件 YSlow，提供了快速分析网站的功能，并且提供了非常实用的优化网站相应速度的建议。当然，我们不会只是针对 YSlow 的推荐进行优化，但是你还是需要安装它。
- **下载安装 Firefox**：http://www.mozilla.org。
- **运行 Firefox 并安装 Firebug**：http://www.getfirebug.com。
- **安装 YSlow:** https://addons.mozilla.org/firefox/addon/5369。
- 马上开始分析你的网站吧，下面是具体的步骤。重启 Firefox 并打开你的网站，选择"工具 >Firebug>Open Firebug"，然后打开"YSlow"标签点击"Performance"。YSlow 会分析你的网站并且给出一个表现等级。接下来看看如何提高网站的得分等级。

6.2　在 IMG 标签中表明正确的图片尺寸

如果有一个 2592×1944 像素的图片，你希望显示为 120×90 像素，那么不要让浏览器对图片进行缩放。

```
<img src="img _ 2592x1944.png" width="120" height="90" alt= "Bad"
/>
```

传输图片文件会消耗很多时间。另外，由于浏览器需要自己将图片缩小，这也会消耗一定的时间而且效果不会太好。所以将图片处理好之后再部署到服务器上，有很多的免费工具可以进行图片的处理，例如 GIMP 和 Paint.NET。

好处：处理过的图片更易于传输，速度更快，提供了更好的用户体验。

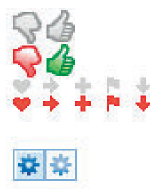
6.3　减小图像的尺寸

应当注意网站中每个图片的尺寸，检查是否可以将图片的调色板设置为 256 色或者更少。当然，这也取决于图像和文件格式。网站中有照片、线图和屏幕截图等，每种类型都需要认真处理。这是一项乏味得几乎没人想从事的工作。

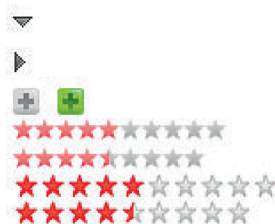
下面提供一个快捷的方法：打开 Firefox 并访问你的网站。在浏览器的右下角，可以看到 Smush.IT 的图标。单击它就会出现一个新的浏览器窗口。这个 Web 服务会获取网站的所有图片，然后进行压缩。下载经过优化的打包 ZIP 文件替换服务器中的文件。同样，在其他有图片的页面中进行相应的处理。

好处：网站中的图片越小，加载时间越短，用户体验越好。

图片聚合



假设你有一个论坛。论坛中的人们喜欢使用表情图标来表示自己的感情。因为表情图标很多，你就需要很多不同的图片文件，这增加了每个页面的 HTTP 请求数。大量的文件下载大大地降低了网站的加载时间。



想象一个 48×16 像素的图片平均分成 3 列，每一列都是 16×16 像素的表情图标：一个是悲伤的表情，一个是高兴的表情，一个是生气的表情，等等。这种单独文件包含了 3 张图片的方式就是图片聚合。使用 CSS 的“background-image” 和“background-position” 属性就可以确定想要显示的图片。例如，在页面上只显示高兴的表情。这样，我们可以将所有的表情都放到一个单独的文件中，极大地减少了 HTTP 的请求次数和时间。自然，图片聚合得越多，你的受益也就越大。

这项技术最早源于计算机游戏行业，适用于一些不经常更新的图片，例如笑

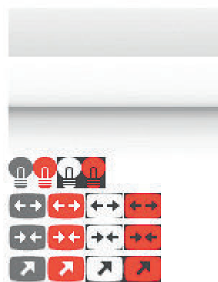
脸、箭头、背景图片以及项目符号。

我们不推荐在菜单图片上使用这种技术，因为图片可能时不时地发生变化，因此每次你必须小心地进行修改，否则很容易出错。仔细的看看 Smashing Magazine 上[1]关于这个话题的讨论吧。

好处：减少 HTTP 的请求数目降低了网站的加载时间，提升了用户体验。

6.4 服务器和内容分发网络

YSlow 建议将静态内容，诸如图片放在世界各地的服务器上，这样图片的存储就更接近用户。如果能够承担服务器的租用费用，那么就请按照这个建议做。但是对大多数用户来说，巨额的费用常常让这个选项不太现实。如果可能的话，租用一台专门的服务器，作为一台廉价的静态内容专用服务器。



Google YouTube 上的图片聚合。

好处：合理使用缓存能够降低贷款费用，提高用户体验。

6.5 合并 CSS 和 JavaScript

通过将所有样式文件放入一个大的样式文件，将所有 JavaScript 文件放入一个 JavaScript 文件，可以减少用户访问网页时的 HTTP 请求数量。当然，开发过程中可以继续使用多样式和多 JavaScript 文件的方法。但是每个开发周期中都需要进行一次文件的整合。

可以通过文本编辑器进行手工的整合。或者，如果你使用 Linux，在 CSS 文件夹中，可以使用 Cat（concatenate 的缩写）命令整合文件。

```
cd your_css_directory
cat file1.css file2.css file3.css > file_to_link.css
```

还有 JavaScript 文件。

[1] http://www.smashingmagazine.com/?p=6319。

```
cd your_javascript_directory
cat file1.js file2.js file3.js > file_to_link.js
```

好处：减少 HTTP 请求数量可以提高网站的响应时间，同时带来用户体验的提高。

因为文件中存在覆盖的问题（CSS 中的属性重写或者 JavaScript 中的代码依赖），在整合过程中需要特别注意 JavaScript 文件和 CSS 文件的加载顺序（你也许会想将多余的 CSS 注释删掉）。如果你决定将文件进行整合，确保在不同文件之间使用注释进行分割，如下所示。

```
/* Menu Styles */
(place menu styles here)
/* Header Styles */
(place menu styles here)
```

6.6 CSS 放在顶部，而 JavaScript 放在底部

正如 YSlow 建议的那样，将 CSS 文件放在 HTML 文件的 <head></head> 标签之间。这样，浏览器在得到 HTML 元素之前就知道该如何进行显示，而不必在加载的过程中对元素进行重新调整。

将 JavaScript 文件放在 <body> 结束标签之前。这样可以避免 JavaScript 对正在加载中的页面进行渲染。

好处：尽可能加快页面的加载速度，提升用户体验。

6.7 压缩 CSS 和 JavaScript

首先，很多基于正则的 JavaScript 压缩器在遇到有问题的代码时非常不稳定。为了节省时间，最好先使用 JSLint[1] 检测一遍代码，将找出的问题分类。即便代码通过了 JSLint 的检测，也应当尽力保证代码接近完美。因为在很多情况下，一个分号都会导致压缩器的失败。

[1] http://www.jslint.com。

在文本编辑器中复制你的 CSS 或 JavaScript 代码，然后粘贴到 Minify[1] 相应的输入框中。整合工具 Packer 3.1 和 Minify 在代码压缩方面有着非常好的表现。单击"Hit Me"，并将结果应用在你的网站上。压缩的代码越多，网站的表现就会越好。对于 JavaScript 来说，效果特别明显，而 CSS 则小不了太多。当然，其他的 CSS 或 JavaScript 压缩工具也许能提供一个更好的压缩比率，你也可以进行尝试。



例如，YUICompressor 是一款集安全、可靠和压缩比率于一体的工具。如果正在使用其他类型的中间件，你也许希望整合 YUICompressor 来实现对 CSS 和 JavaScript 的自动处理。作为一个 Java 程序，YUICompressor 可以整合在任何地方。

好处：减小了文件尺寸，降低了带宽消耗，JavaScript 的表现得到了显著地提升，改善了用户体验。

6.8　使用外部 CSS 和 JavaScript 文件

不要使用内联的 CSS 或者 JavaScript，把它们都放在外部单独的文件中。这样不仅易于理解（需要进行修改的时候，你知道去修改哪个对应的文件），而且也减少了需要缓存的 HTML 文件的尺寸，提高了缓存的利用率（例如，在一个 JavaScript 或 CSS 文件被多个页面使用的时候），同样也降低了 HTTP 的请求数。

好处：改善了缓存利用率，页面重新加载时间变得更短，提升了用户体验。

6.9　选择一个文档类型

选择文档类型虽然对于改善速度没什么直接影响，但是通过为浏览器指定清

[2] http://www.reneschmidt.de/tools/minify/。

晰、明确的 HTML 和 CSS 解析规则，可以充分发挥它们的潜力。

你在考虑在网站中使用 HTML 还是 XHTML 标准？ HTML4.0 是一个很好的选择，尽管仍有不少的站长乐于选择 XHTML1.0。不要在任何地方漏掉文档类型的声明，这样会导致浏览器去猜测应该使用什么标准。相反，在 HTML 的第一行就声明文档类型。

下面是 XHTML1.0 Trasitional 的例子。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
```

下面是 HTML4.01 Transitional 的例子。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
```

好处：告诉浏览器处理网站的方式，是非常重要的步骤，确保了在不同浏览器中网站外观的一致性。

6.10 验证页面

特别是在你提供基于网络的服务时，潜在用户会仔细分析你的网站，来判断你对于质量的关注究竟到何种程度。通过 W3C[1] 的验证服务页面表明你对于质量非常关注，而且致力于设计在普通网络浏览器上表现一致的页面。经过验证的页面是否代表了网站的服务质量是另外一回事儿，但是如果潜在用户在你的网站上验证失败，他就会对你失去信任。

6.11 不要使用 CSS 表达式

支持 CSS 表达式的浏览器允许你将逻辑放在 CSS 的声明中。这种做法既不被

[1] http://validator.w3.org。

网页标准所接受（因此很多浏览器并不支持这个特性），也不是一个好习惯，它违背了 CSS 将程序逻辑和内容表现分开的初衷。例如，当 JavaScript 脚本或窗口缩放触发了文档中对象的移动或更新时，对象的样式也需要相应更新。CSS 表达式要比普通的 CSS 花费更多的步骤，处理速度明显地降低了，这是一种资源的浪费。千万不要使用 CSS 表达式，它们一无是处。它们从来没有被网页设计社区所接受，即便是它的“创造者”微软，也在 IE8 中抛弃了它。

6.12 服务器部分

接下来是一些较为专业的内容。你应该已经拥有了服务器的 root 权限。也许因为使用了共享主机，你无法获得这样的权限，但这部分仍然会对你有用。你可以检查自己的主机是否满足需求，如果不能满足，也知道该如何去提出你的需求。

为了掌握这些信息，你也许需要时不时地看看服务器文档。

配置 Web 服务器需要一些服务器的底层进入权限，通常是通过 SSH 的方式。你应当具有一些 SSH 的基本常识，例如如何登录、如何作为 root 用户修改配置文件以及如何重启服务等。每种服务器软件都不一样，所以本节不可能介绍所有的具体操作细节。使用 Debian GNU、Linux、Windows、Red Hat GUN/Linux 等不同的环境，具体的步骤是不相同的。

本节中使用标准的 Debian GNU/Linux（代号“Lenny”）操作系统。

6.13 Apache：如何设置过期时间

HTTP 标头中的“expires header”属性决定了文件多长时间应当进行刷新。文件被缓存后，只有在过期的情况下浏览器才会重新发送请求，这样就减少了 HTTP 的请求次数。

我们开始吧。首先通过 SSH 登录服务器并且获得 root 权限。

```
su
```

如果你想安装一个容易使用的文本编辑器，在命令行中执行下面命令。

```
aptitude install nano
```

继续，打开 Apache 的过期模块。

```
a2enmod expires
```

确保 .htaccess 文件的解析功能已经打开。编辑网站的配置文件，配置文件名变化很大。也许存放在“/etc/apaches/sites-enables”下，假设文件名为“ooo-default”并且使用“nano”编辑器进行编辑。

```
nano /etc/apache2/sites-enabled/000-default
```

如果“AllowOverride”属性已经被设置为“All”，那么就这样退出，否则修改“AllowOverride none”为“AllowOverride All”，保存退出，然后检查 Apache 的配置文件。

```
apache2ctl -t
```

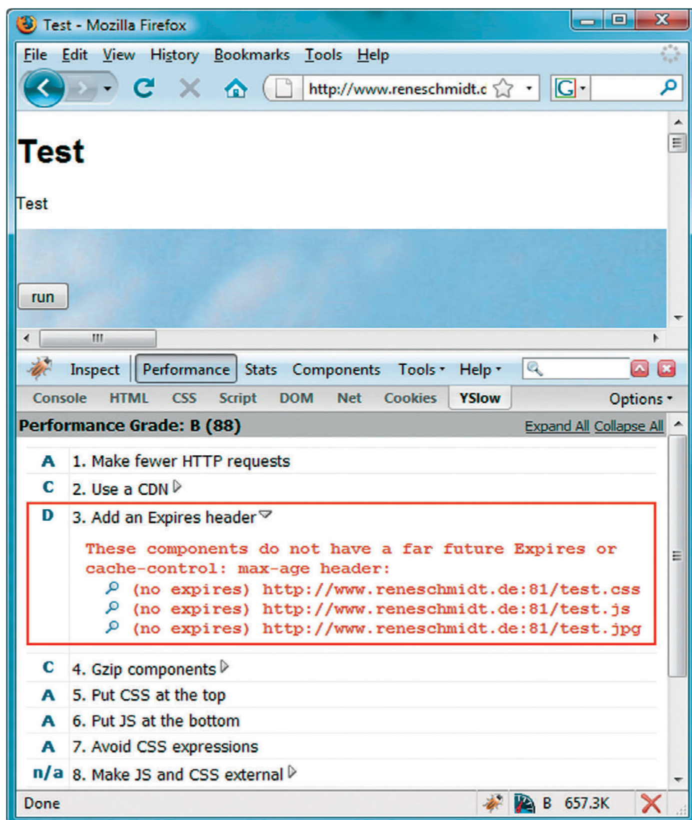
如果一切正常，那么重启服务并退出。

```
/etc/init.d/apache2 restart && exit
```

切换到 FTP 或者 SCP 客户端，登录到网站的 wwwroot 目录并创建一个“.htaccess”文件。注意前面的点，用你的文本编辑器打开，并且输入下列内容。

```
ExpiresActive On
ExpiresDefault "access plus 1 year"
ExpiresByType text/html "access plus 1 hour"
```

这个设置让 HTML 缓存一个小时，其他的内容缓存一年。经验之谈：让静态文件缓存更长的时间。你可以实验为不同的内容设置不同的过期时间，例如“text/javascript”、“application/javascript”、“text/css”、“image/gif”、“image/png”和“image/jpg”等。



使用 YSlow 检查刚才的配置是否生效。首先清除浏览器的缓存。

6.14 YSlow 中可以查看缺少 expires header 的报警

如果设置了 expires header，即使你更新了 CSS 或者 JavaScript 文件，用户访问的可能仍然是旧的缓存文件。在地址中加入版本参数可以简单地消除这个问题，如下所示。

```
<link rel="stylesheet" type="text/css" href="style.css?v= 1.0"
/>
```

当然，这个参数并不会被接受或处理。它仅仅是给浏览器的一个指示。当文件更新的时候，将更新这个参数，例如从"v=1.0"到"v=1.1"，这样浏览器

就会取回更新后的内容。

6.15　Lighttpd：如何设置过期头

Lighttpd 或者 Lighty 并不读取 .htaccess 文件，所以我们必须把设置写在它的配置文件中。切换到 root 用户进入 /etc/lighttpd/conf-available。

```
su
cd /etc/lighttpd/conf-available
```

创建一个这个模块的配置文件。

```
nano ./01-expire.conf
```

为简单起见，我们为所有的虚拟主机（vhosts）设置相同的过期头。把下面的内容放入新创建的文件中。

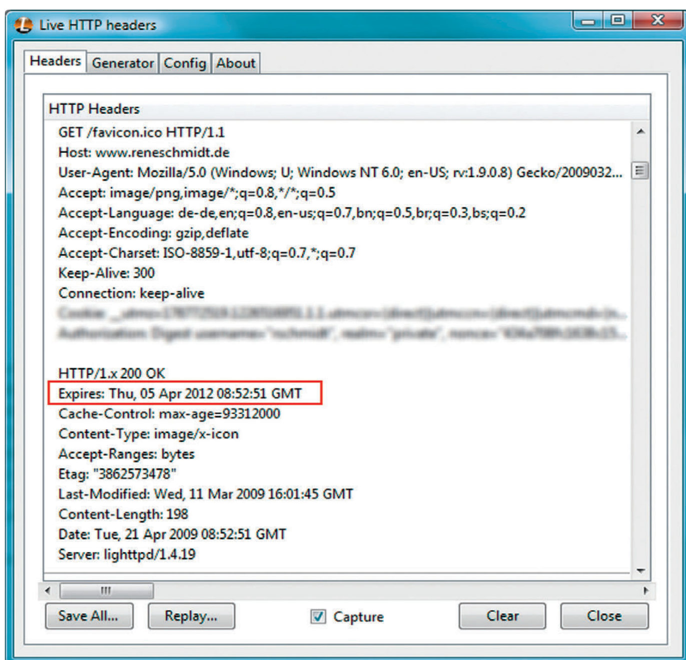
```
server.modules += ( "mod_expire" )
$HTTP[ “url” ] =~ "" {
expire.url = ( "" => “access 12 hours" ) }
$HTTP["url"] =~ "(7z|rar|zip|gz|pdf|exe|css|js|png|gif|
jpg|ico)$" {
expire.url = ( "" => "access 1 years" )
}
```

这样静态文件的过期时间被设置为一年，其他访问的内容过期时间为一个小时。保存文件并退出，把新文件与当前的配置文件合并，并且重启 Lighty。

```
lighttpd-enable-mod expire
/etc/init.d/lighttpd force-reload
```

6.16　Apache：如何配置 ETags

ETag（实体标签的缩写，entity tag）是 HTTP 响应头（response header）中用来判断文件内容是否被修改的标志。当 HTTP 响应中包含与之前相同的 ETag 时，请求的内容被认为没有修改过进而返回空的响应体。



现在 Lighty 输出了过期时间。

在 Apache 中，文件实体标签默认为“All”。表示文件的 ETag 是由 Inode、Mtime 和文件尺寸信息生成的。如果你的服务器不支持 ETags，可以手动地启用它。编辑网站中的 .htaccess 文件，查找“FileETag”，也许会找到“FileETag None”，修改如下所示。

```
FileETag All
```

保存并上传文件，然后检查现在 Apache 是否提供了 ETags。如果不支持，可能是因为 Apache 编译的时候选择了禁用 ETags。那么可以换用另外版本的 Apache 或者请求主机提供商启用这个特性。

6.17 Lighttpd：如何配置 ETags

Debian 中的 Lighty 默认开启了 ETags，所以基本上不需要操作。如果没有开启，可以通过下面的命令打开，切换到 root。

```
su
cd /etc/lighttpd/conf-available
```

创建新的配置文件。

```
nano ./10-etags.conf
```

将下面的代码添加到文件中。

```
static-file.etags = !"enable"
```

保存退出。加载新的配置文件，重启 Lighty。

```
lighttpd-enable-mod etags
/etc/init.d/lighttpd force-reload
```

6.18 为 PHP 安装 eAccelerator

Debian 没有预置的 eAccelerator 包，所以需要从源码进行编译。假设服务器中已经安装了 Debian Lenny 和 PHP5，首先下载一些包。

```
aptitude install bzip2 php5-dev make
```

切换到 root，从 eAccelerator 网站下载源码。

```
su
cd /tmp
wget http://bart.eaccelerator.net/source\
/0.9.5.3/eaccelerator-0.9.5.3.tar.bz2
```

文中使用的版本为 0.9.5.3，如果想使用最新的版本，替换相应的版本字符串。

解压源码并且切换目录。

```
tar xvjf eaccelerator-0.9.5.3.tar.bz2
cd eaccelerator-0.9.5.3
```

开始源码编译。

```
export PHP_PREFIX="/usr"
$PHP_PREFIX/bin/phpize
```

```
./configure -enable-eaccelerator=shared \
-with-php-config=$PHP_PREFIX/bin/php-config
make && make install
```

eAccelerator 的配置文件需要手工创建。

```
nano /etc/php5/conf.d/ea.ini
```

将下列内容复制至配置文件中。

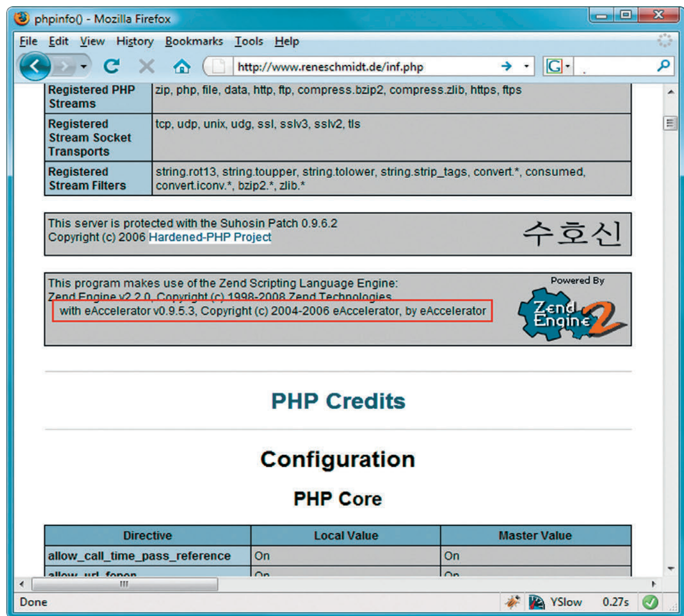
```
[eAccelerator]
extension="eaccelerator.so"
eaccelerator.shm_size="32"
eaccelerator.cache_dir="/tmp"
eaccelerator.enable="1"
eaccelerator.optimizer="1"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
eaccelerator.filter=""
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```

重启服务器。

```
/etc/init.d/lighttpd restart
```

或者

```
/etc/init.d/apache2 restart
```



正确安装 eAccelerator 后，红框的部分会出现一个额外的版权信息。

检查 eAccelerator 是否编译并安装成功。切换到 FTP 或者 SCP 客户端，进入网站的根目录，创建一个“inf.php”文件，添加下面的代码。

```
<?php phpinfo(); ?>
```

保存退出并上传文件，在浏览器中访问这个文件。

```
http://www.example.com/inf.php
```

当然，你需要将 www.example.com 替换为实际的主机名。别忘了将服务器不需要的包删除。

```
aptitude remove php5-dev make
```

重要提示：移走具有公共访问权限的 inf.php 文件。如果黑客能够访问它，就能得到很多有助于攻破系统的有用信息，任何人都不希望遇到这样的安全问题。

6.19 MySQL Tuning Primer

同样，调试 MySQL 也需要 SSH 权限。还需要一个能获得 MySQL 运行信息的账户（例如 root,但是同系统的 root 用户不同）和一个运行的 MySQL 服务器。登录服务器并且下载 MySQL Tuning Primer。

```
wget http://launchpad.net/mysql-tuning-primer/trunk\
/1.5-r1/+download/tuning-primer.sh
```

赋予执行权限。

```
chmod +x ./tuning-primer.sh
```

另外需要解决这个脚本的依赖关系，该脚本依赖于“bc”一个很小的命令行计算器。

```
aptitude install bc
```

运行脚本，按照提示输入 MySQL 账号信息，脚本会分析运行状态信息并且为 MySQL 的配置优化提供建议。仔细阅读提供的建议，对 MySQL 的配置进行相应的修改。

```
nano /etc/mysql/my.conf
```

例如，如果 Tuning Primer 提示“你的 query_cache_size 太高，也许你可以将资源用在其他的地方”，那么根据提示修改配置文件，查找“query_cache_size”，大概降低 10%。一次不要修改太多，保持在 10% ~ 20%。保存退出后重启 MySQL。

```
/etc/init.d/mysql restart
```

建议在进行下一次调试前等待 48 个小时，以让 MySQL 收集足够多有用的信息。

6.20 启用透明压缩

如果你的网站访问量不大，只是提供一些小的 HTML、CSS 和 JavaScript 文件，压缩文件并不是十分必要。因为即便负载高到严峻的程度，即时的压缩会给服务器带来额外的压力。

但是如果你有大量的资源和大尺寸文本文件（例如 HTML、CSS、JavaScript 和 JSON），此时应当激活压缩功能，这样可以节省带宽、减少传输时间。

6.20.1　启用 Apache 的压缩功能

下面是为 Apache 启用压缩的方法，切换到 root 用户，编辑 deflate.conf。

```
su
nano /etc/apache2/mods-available/deflate.conf
```

默认情况下，Debian Lenny 中 Apache 的 mod_defalte 模块默认对“text/html”、“text/plain”和“text/cml“文件进行压缩。这还不够，也许我们希望 CSS、JavaScript 或者 JSON、ATOM 和 RSS 也进行压缩，那么将 deflate.conf 修改如下所示。

```
<IfModule mod_deflate.c>
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE text/javascript
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE application/javascript
AddOutputFilterByType DEFLATE application/
x-javascript
AddOutputFilterByType DEFLATE application/json
AddOutputFilterByType DEFLATE text/json
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/rss+xml
AddOutputFilterByType DEFLATE application/rdf+xml
AddOutputFilterByType DEFLATE application/atom+xml
# Netscape 4.x has some problems...
```

```
BrowserMatch ^Mozilla/4 gzip-only-text/html
# Netscape 4.06-4.08 have some more problems
BrowserMatch ^Mozilla/4\.0[678] no-gzip
# MSIE masquerades as Netscape, but it is fine
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
</IfModule>
```

注释行在官方文档中已经被去掉。

由于浏览器不请求或接受编码为图片的内容，有时我们会尝试为所有的文件设置压缩。但是这么做的话，在浏览器请求已经被压缩过的较大文件时会发生严重的问题，很可能会导致类似 DDos 的情况。因为图片、视频或者类似类型的文件已经被压缩过，所以我们无需再压缩这些文件。即使浏览器要求压缩这些内容，服务器也不会进行压缩。

6.20.2 激活 Lighttpd 的压缩

Debian Lenny 中的 Lightpd 默认打开了压缩的选项，所以不需要进行手工的设置。即便如此，最好知道如何设置，也许今后会派上用场。

切换到 root 用户，进入 /etc/lighttpd/conf-available。

```
su
cd /etc/lighttpd/conf-available
```

创建新的配置文件。

```
nano ./05-compress.conf
```

将下面的代码加入新的配置文件中。

```
server.modules += ( "mod_compress" )
```

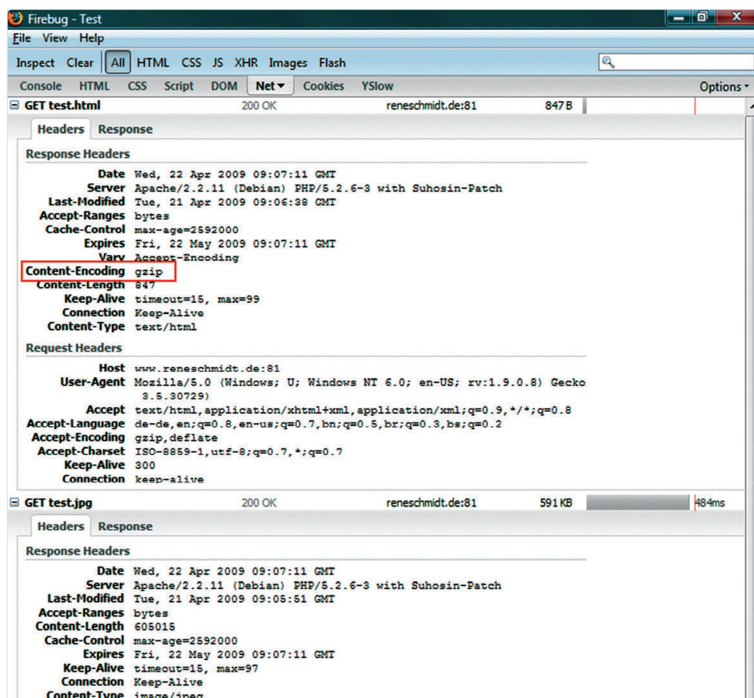
保存退出后整合配置文件，重启 Lighty。

```
lighttpd-enable-mod compress
/etc/init.d/lighttpd force-reload
```

6.20.3 打开 PHP 的压缩选项

你也许想激活压缩输出的 PHP 脚本。使用“su”命令切换到 root 账号，编辑 php.ini 文件。php.ini 的路径取决于使用的服务器和 PHP 的安装方式。对于 Apache 和 mod_php 而言，地址如下所示。

```
nano /etc/php5/apache2/php.ini
```



已经被压缩的文件不能被再次压缩。

以 FastCGI 方式安装的 Lighttpd，配置如下所示。

```
nano /etc/php5/cgi/php.ini
```

在配置文件中查找“zlib.output_compression”打开这个选项。

```
zlib.output_compression = On
```

保存退出并重启服务器。

```
/etc/init.d/lighttpd restart
```

或者

```
/etc/init.d/apache2 restart
```

6.20.4 检查压缩效果

重新打开 Firebug，切换到 Net 标签，并且重新加载你的网站。现在 HTML、JavaScript、CSS、XML、RSS、ATOM、JSON 和普通的文本内容应当默认会被压缩，其他类型的文件则不会被压缩。

6.21 尽可能避免重定向

网站在使用一段时间后会产生很多重定向。有些文件因为过期或者突然消失而被移走了，当用户访问这些文件时对请求进行重定向是一个很好的办法，这样可以帮助用户找到他搜索的内容。如果要使用重定向，首先检查重定向是否已经存在了。

```
A  -> B  ->  C -> D
```

在这个例子中，用户请求文件 A，接着请求被重定向到 B，然后重定向到 C，最后重定向到 D。整个过程有四次请求，只有两次请求满足了。

```
A -> D
```

好处：如果能够仔细的优化重定向链，服务器就能够处理更多的并发访问。

6.22 如何探测长时间跳转链?

切换到 Firefox，打开 Firebug，点击 Net 标签。如果被禁用，那么选择“Network monitoring”并且应用设置。这样 Firebug 会重新加载当前的页面，并且列出加载页面所有必要的请求。

当用户请求一个 HTML 文件的时候，Firebug 会列出这个请求同时发生的所有请求以及请求的状态码。302 表示的是一个临时的重定向。示例中的服务器并没有进行过优化，所以不能同生产环境的服务器进行对比。在

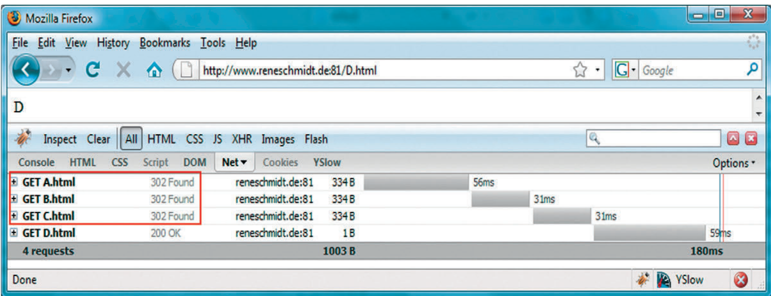
上例中，四个请求耗费了 180ms（毫秒）。如果页面的请求导向了一个长时间的跳转链，响应时间就会超过 1s（秒）。你应该让用户直接访问实际的地址，如下所示。

```
Redirect /A.html http://www.example.com/D.html
```

而不是进行间接的重定向（在 .htaccess 中设置），如下所示。

```
Redirect /A.html http://www.example.com/B.html
Redirect /B.html http://www.example.com/C.html
Redirect /C.html http://www.example.com/D.html
```

短的重定向涟处理起来比长链快得多。本例中，重定向消耗了 105 微秒。



避免不必要的重定向。

6.23 完整的链接

打开永久链接功能的 WordPress 会自动重定向到一个不带结尾斜线的链接。

```
http://www.example.com/post -> http://www.example.com/post/
```

注意结尾的斜线，这产生了两次请求，其中一次没有生效。那么，务必确保 WordPress 的永久链接带着结尾的斜线，这样可以减少必要的请求数。换句话说，如果你要添加内部的链接，确保这些链接总是带着结尾的斜线，除非是链接到像 *.html 或者 *.php 这样的静态文件。

另外，同样的内容不要使用两个链接。确保指向 http://www.example.com/postname 的请求确实重定向到 http://www.example.com/postname ，这个

规则不是为了性能的目的，而是为了优化搜索引擎。因为相同的内容会对排名造成影响。

6.24 避免死链接

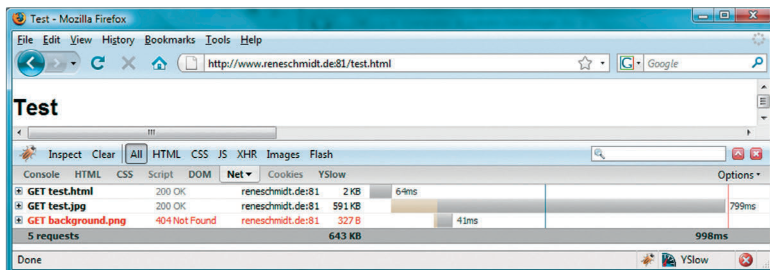
请注意，如果请求了一个不存在的文件，服务器必须发送一个 404 信息明确地告诉用户请求的文件找不到了。作为站长，你应该确保没有链接到不存在的 HTML、CSS 和 JavaScript 文件。很多服务器在单独的文件中记载 404 日志，众多的 404 请求会不断增加日志文件的大小，而且增加了服务器的资源消耗（例如 CPU 和磁盘空间）。

如何检查内部的死链接?

检查内部的死链接就像是在公园中的漫步。假设网站中有一个几乎全白的背景图“background.png”，通过“background-image”属性赋给了 body 标签。

```
body { background-image: url("background.png"); }
```

假设这个文件不在了，仅仅浏览网页，你还无法发现这个问题。浏览器也不会明确地报告通过 CSS 文件加载文件的问题。现在打开 Firefox 并且清除浏览器的缓存。打开 Firebug，重新加载当前的页面。Firebug 会列出所有的请求，包括 CSS 文件中包含的文件。



Firebug 会将 404 的请求高亮显示出来。

Firebug 高亮显示了 404 结果的请求。通过将文件放到正确的位置或者修改 CSS 属性，来修改这个问题。这样，你就能更有效率地使用服务器的资源。

Page Speed 是 Google 内部已经在使用的一款提高网络应用程序性能的工具。像 YSlow 一样，它是一款 Firebug 的插件。

虽然有些相似，但 Page Speed 还提供了一些 YSLow 没有的特性。

- JavaScript 事件
- 显示没有使用的 CSS 样式
- 显示效率不高的 CSS 选择器

网络应用能够从如此精细的优化中获益匪浅，而这个插件显然非常适合大型网络应用。作为一名网络应用开发人员，为什么不试一下呢！

第7章

为销售而设计——提高访客转化率

Dmitry Fadeyev

每个网站的站长都希望网站表现很好，获得更多的销售额、用户注册或者 RSS 订阅者。那么如何将一个新的访客转化成为忠实的客户呢？本章将着眼于有效的销售理论并提供一些提高访客转化率的特殊技巧。

大多数漂亮或者给人印象深刻的网站并不仅仅是件艺术品，它们的功能界面都服务于一定的目的。如果你经营一家在线商店，你的目的就是卖出更多的商品。如果您运行了 Web 应用程序，您希望更多的用户来注册。不管您在什么行业、经营什么样的商业、组织或者社区，您都希望通过网站来提高销售额、注册量、RSS 订阅数或者点击量。

多数站长都希望有一个很高的转化率。但是，怎么样才能把新访客转化成忠实客户呢？

“转化”是在线营销的一个术语，描述了网站访客做出站长期望的动作这一过程。例如，如果你经营一家在线商店，销售一件商品就是一次转化；如果是博客，增加一个订阅就是一次转化。转化的效率用网站的全部访客与进行了特定动作的访客之比——转化率来表示。

为了解答这个问题，我们先来看看如何进行有效的销售。

7.1 如何销售？

要更有效地销售，你必须销售解决方案，而不仅仅是产品。还需要让用户受益，而不仅仅只是提供产品特性。用户并不是需要产品、服务或是特性，他们需要解决自己问题的整体方案。特性是产品或者服务外在的修饰品，而受益是用户使用产品过程中获得的，促使他们最终购买产品的原因。例如，对顾客来说，iPod Nano 有 16GB 的空间只是说明了产品的技术细节但是没有说明它能给用户带来什么。假如说 iPod Nano 可以容纳 4000 首歌曲则明确地表明了高存储给用户带来的好处。这恰好解决了用户希望将整个音乐库装

入口袋的需求。

这些障碍正是客户不购买产品的原因。这些障碍或大或小，也许他们讨厌你的产品或者只是没有购买的预算，也许他们没有看到自己需要的特性或者认为太贵或者他们不喜欢这个颜色。

一旦你了解了你正在出售的产品（对于产品的解决方案和带给客户的好处了如指掌），你就需要去打破客户评估你的商品时产品和客户之间的障碍。

这些障碍有希望被打破和处理。例如，如果潜在客户觉得产品价格太贵，那么就给客户指出他花这笔钱之后，将会为他节省多少时间和精力。

如果产品缺乏特性，那就可以谈一些通用的解决方法，毕竟，客户其实只是在为他们的问题寻找解决方案而不是十分关心产品特性。如果你能说服客户，你的产品能够解决他的问题（也许和客户心目中的方案不太一样），你就可以成功。

7.2 AIDA（注意力、兴趣、期望、行动）

AIDA 代表了注意力、兴趣、期望、行动，是一个非常流行的销售方法。AIDA 是为销售广告确定基调的指导。任何销售广告的目标都是完成交易，任何方法都应该力争做到效果最大化。到最后销售时，目标客户已经非常希望购买商品了。像在标准的商业广告中一样，AIDA 也可以用于客户停在网站的这段时间，这个时候面向用户的不是销售人员，而是我们的网站。

AIDA 广告的第一阶段是“吸引注意”。这一点非常重要，因为潜在客户可能会飞快地浏览你的网站然后离开。你只有一段很短的时间来抓住客户注意力，并且促成一桩交易。这段时间绝对是非常重要的，因为访客离开后，你所做的一切都是徒劳的。一定要通过你的解决方案能给客户带来的生活改变来吸引客户。简洁、有力、清晰地告诉用户关注我们的原因。

接下来两个阶段，“兴趣”和“购买欲望”是销售产品的环节。如果你在“注意力”的阶段做了足够的文章，现在就是巩固的时候了。清晰、简洁地介绍产品用途以及带给用户的真实价值。不断传达给访客能享受的收益来激起他

的购买欲望。如果你为访客描绘了一幅栩栩如生的画面，让用户觉得使用你的产品可以提供足够多的受益，那么他们的购买欲望就会膨胀起来。

来自 37Signals 的网站 The Highrise 使用了 AIDA 的方法来延长会话时间。首先，网站通过非常精炼的产品概述和让客户受益来抓住访客的注意力，“与商业伙伴保持沟通的好方法”。然后通过一个给用户带来好处的列表吸引了访客的兴趣。那些谈论 Highrise 如何帮助商业公司的视频引起了访客的购买欲望。最后，一个很大的按钮诱使你去看看计划的价格列表。

最后一步就是“行动”。如果我们是在销售有形的产品，可以邀请用户购买并且告知用户如何付款。如果你的产品是网络应用或者服务，我们就需要吸引访客去注册。在这个阶段，我们需要终结一次交易，所以将访客的注意力吸引到行动上是非常重要的。那些想购买的人们，只需得到最后的如何付款或者注册的指引，那些还在观望的人们则需要我们最后再推一把。

最近，代表“满意”的 S 加入到 AIDA 中来。能够得到用户已经很好，如果再能让客户将你的产品推荐给他的朋友，那就更棒了。这样你就能确信客户对服务非常满意。我们不需要将产品做到完美，但是我们应该为用户提供一流的服务。解决用户服务问题是培养忠实用户的好方法，这表现在客户遇到突发问题时公司仍是客户最可靠的伙伴。



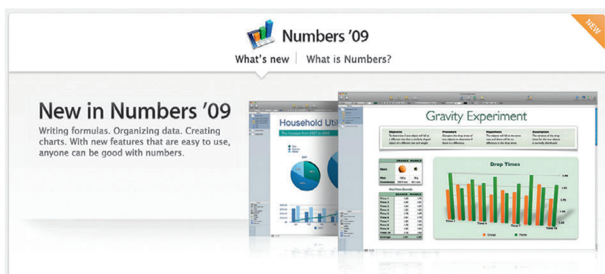
AIDA 销售策略

7.3　展示产品

如果在实体商店购买产品，不但可以看到还可以亲自体验这些产品。在购买之前能够看到产品为客户提供的更多信息。你对将要购买的产品也会充满好感。

但是当客户在线购物时，我们能够得到的信息类型就非常有限，而有的网站甚至连产品的外观照片都没有。

即使是销售数字产品，例如桌面或者网络应用程序，展示一些截图或者视频是非常必要的。访客看到产品的外观后，他们就可以想象实际使用的情形。对用户来讲，这非常重要。如果能够让潜在客户想象使用产品的情形，你就已经开始勾起他们的购买欲望了。更重要的是，好看的外观不仅仅会吸引访客，还传递给访客易于使用的信息[1]。



Apple 公司在页面顶部为他的图表应用 Numbers 放了很多大截图。

[1] “Apparent Usability vs. Inherent Usability: Experimental Analysis on the Determinants of the Apparent Usability” ,by Masaaki Kurosu and Kaori Kashimura, CHI’ 95 Conference Companion, 1995, p.292-293。

7.4 使用视频展示产品

视频正逐渐成为在线展示产品的一种极受欢迎的媒介，特别是对于像桌面或者网络应用程序这样的数字产品。这是因为在一个视频短片中你能提供比文字和截图更多的内容。对访客来说，视频只要看和听就可以了，这对他们来说更容易理解。

使用视频展示的最好方法是将视频放在登录页的显著位置。如果事先能够准备一份有声的底稿放在视频的顶部，在声音不清楚的时候避免访客发生误解。底稿应该保持简洁并且切中要点。

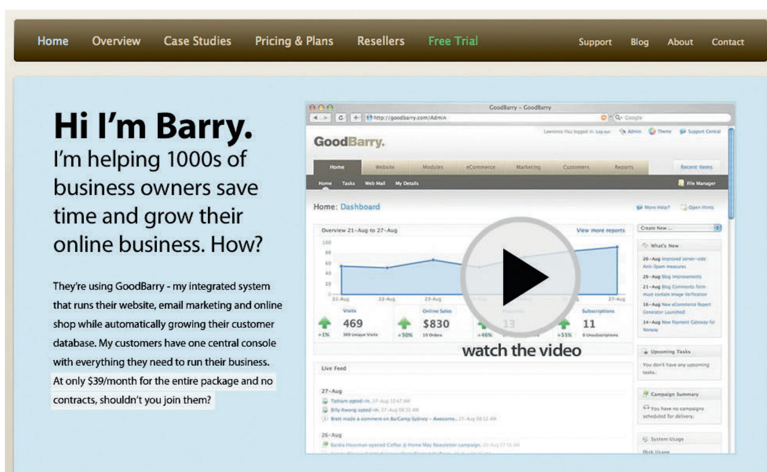
视频中非常适合 AIDA 的结构，首先你可以谈一下产品能够解决什么问题，产品是如何解决问题的，能够给客户带来什么好处，接下来你就要邀请用户注册或者继续浏览网站来获得更多信息了。

7.5 可以浏览的功能列表

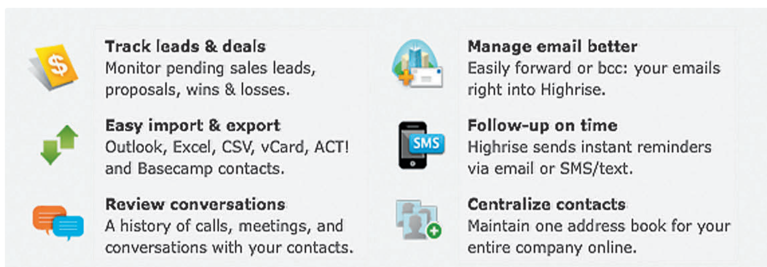
如果你正在销售一件产品或一项服务，肯定会有一堆的功能要去宣传。你非常希望人们能够看到并且理解这些功能。但列表往往是单调的，特别是列表中包含了过多文字的时候。人们浏览网站时一定不会花太多精力去读你的广告，因为这不是在工作，所以你要想办法把列表尽力做得简明易懂。幸运的是，已经有很多现成的方法可供我们参考。

首先使用图标或图片。功能描述列表旁边的小图标就像是项目符号。我们可以很容易地注意到这些分布在页面上的小锚点。图标或图片比文字更有趣、更容易吸引眼球。

为每个功能使用短标题，一般是很少的几个词或短句，能够和图标一起一览而过。为了吸引访客注意，字体要比描述文本大，而且和背景的对比度也要强烈一些。短标题下方应当是稍微详细一点的描述，描述的字体要小一些、对比度要弱一些。这部分的元素应该淡入背景，如果访客喜欢这个题目，他们会仔细阅读这些内容，同时描述部分不会妨碍访客浏览整个列表。最后，在每个项目之间添加足够的空白更有助于浏览。



GoodBarry 的登录页嵌入了一个很大的截图视频。巨大的播放按钮吸引了用户注意，促使他们点击它。



Highrise 网站上的列表综合使用了小图标、对比度和空白，整个列表变得非常容易浏览。另外，每行中还有能够带给访客更多有用信息的描述文字。

7.6　馈赠让客户想回报

另一种将你带到销售或者谈话中的技巧，就是免费赠送。当你收到一份礼物，你会有种想要报达的感觉，同样回馈一些东西。Robert Cialdini 是阐述影响力方面的最好的作者之一，在《影响力：科学与实践[1]》一书中分享了一些如何利用赠品帮助销售的案例。如果从陌生人那里接收了一个小礼物，人们

[1] Cialdini,R., Influence:Science and Practice, 5th ed.,(Allyn & Bacon 2008)。

从他们那里购买奖券的几率就增加了一倍。付给五美元填写一份问卷和完成一份问卷后付给五十美元比起来，客户更倾向于立刻获得五美元。在饭店中，随账单附赠一些糖果能够让顾客为你提出更有价值的建议。



Linkscape
A professional quality inlink tool that uses patented SEOmoz metrics. Inlinks, anchor text distribution and more. Linkscape's metrics are also available in the Site Intelligence Service API.



Trifecta
Measures metrics to estimate the relative popularity and importance of Page, Blog or Domain. Trifecta replaces SEOmoz's Page Strength tool.



Term Target
Helps determine how targeted a particular page is for a specified keyword by analyzing a variety of factors.



Backlink Analysis Tool
Get an advanced look at the keywords websites are linking to you with.



SEOmoz Firefox Toolbar
The power of Linkscape is now available inside Firefox with the SEOmoz Toolbar.



SEOmoz Labs
SEOmoz Labs is a place where our more adventurous PRO users can check out the *bleeding edge* of SEOmoz technology and product design.

SEOmoz 为专业的 SEO 人士提供商业工具，同时也提供很多免费的小工具。这是一个吸引人气和投资回报的好方法。

那么，在你的网站上，你要送出什么礼品才能让用户觉得自己需要去回报呢？如果销售软件，尝试一下赠送 30 天的免费试用资格？或许可以通过将产品的一部分功能完全免费发放，而采取“增值服务”模式，即免费使用基本功能而需要为额外的功能付费？一些网站制作了很多小工具免费发放给用户。例如，37signals 提供了 Tadalist 和 Writeboard 的免费工具，这些是其商业版本的一些附加组件。通过为客户提供一些免费的应用，客户就更加倾向于先试用，然后购买其他部分的功能。

7.7 从众心理

从众心理是一种心理学现象。当人们不确定应该采用什么步骤时，他们通常会选择同其他人一样。例如，一个著名实验，一些人站在原地仰望天空，很多路人注意到后也开始仰望天空，想看到其他人在看什么。实验证明，从众

心理对人的影响力非常大，以至于到最后为了避免堵塞交通不得不终止。

人们跟随别人的时候觉得更加放心。如果他们看到其他人买了一件商品，他们会放心地去买，因为他们知道这件商品很流行。这非常好，不是吗？我们可以利用从众心理来帮助提高产品销量或服务的转化率，大量地进行广告宣传，将产品的流行性以及已往客户的好评告知潜在客户。例如，用户评论是推荐产品的好方法。Amazon 做过类似的事情，其在页面上嵌入了一个“买过这件商品的用户还购买过”的版块，中间展示了基于其他用户购买行为筛选出来的商品。



Amazon 推荐的购买相同产品的客户还购买了的商品。Amazon 还允许用户对产品进行评论，很好地利用了从众心理现象。

7.8　隐性的建议

图片和组件对用户的影响也起到重要作用。研究表明，给一个孩子展示一个圣诞老人的帽子可以增加他们与其他人分享糖果的可能性，而向他展示 Toys ‘R’ Us（玩具反斗城）的标志则会起到相反的作用。商家经常会将某种图片和标志与商品之间建立联系，当向用户展示精心挑选的图片时，用户就会置身于这种环境，并作出商家期望的动作。

在网站设计中可以使用这个办法。不要仅仅为了装饰而选用图片，要仔细考虑你希望给访客什么样的情感体验。你想从图片传达什么信息给用户？选择那些有意义的图片，这样当他们看到图片时，脑中就会定格在与产品相关的那一幅画面。

7.9　永远提供下一步操作

什么时候应该结束推销？这个问题的答案很简单：“永远不要结束推销”。你将在不同的阶段、以不同的速度将访客转化为客户。所以继续推销，你可

以将那些已经打定主意的访客拉到我们的营销中来，他们所要做的只是最后的一步——注册。在网站的每个页面上都提供购买、注册或者订阅的链接，有的页面上也许可以出现多次。当然，不要在页面中过度地堆积这些操作，而应适度地提供一些捷径，让准备好注册的人们随时可以点击。



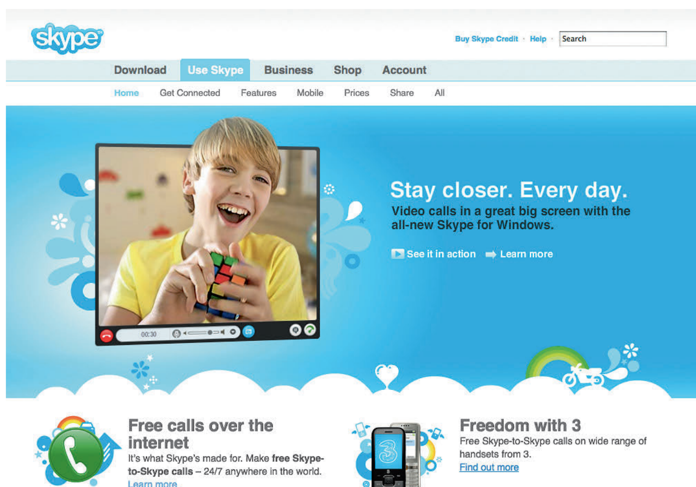
Legacy Locker 展示了一个幸福的家庭照片，旁边并附有介绍，以引起访问者注意，去照顾和保护他们的亲人。

我们需要致力于保持流程的畅通，也就是说不能有死胡同。例如，当访客浏览到营销广告或宣传信息的底部时，下一步要去哪儿？在类似版块或者子版块的位置提供一个到下一个版块的链接。通过引导，而不是让访客自己去寻找，来帮助他们更容易地浏览我们的广告。可以通过在简要介绍底部添加“了解更多”、“立即购买”或“浏览网站”标签，确保访客总是知道下一步要做什么，当他们浏览完一个章节，一个链接就像在等待指引他们进行下一个阶段。

7.10　使用稀缺性来促进需求

当限量版的商品开始售卖时，能够拥有它会带给他的主人一种优越感。因为这类商品一旦售出，再想购买就非常困难了，因此商品的价值就提升了。Google 第一次推出自己的电子邮件应用 Gmail 的时候，就利用了稀缺性。他们通过限制使用权限来创造稀缺性，你只有被已经拥有账号的人邀请才能使

用。能够拥有一个 Gamil 变成了一种特权，人们比免费开放注册的时候更希望得到一个 Gmail 账号。



Skype 页面中没有死链接。注意，每个功能描述下方都是一个或多个下一步的链接，比如“了解更多”或者“马上体验”。

稀缺性还会让访客认为如果他们动作慢了，很可能就会错过这次购买的机会，因此他们会很快行动。对限量供应或库存有限的产品进行广告宣传是个很好的办法，那些本来无意的人们会马上购买或注册。

Product	Brand	Stock	Price	Qty	
Intel X25-E Extreme 32GB 2.5" SATA-II Solid State Hard Drive (SSDSA2SH032G1) The Intel Extreme SATA-II Solid-State Drive (SSD) offers outstanding performance and reliability for servers, storage, and high-end gaming systems. ☆☆☆☆☆	Intel	10+ in stock	£324.99 (£282.60)	1	Buy ▶
Intel X25-E Extreme 64GB 2.5" SATA-II Solid State Hard Drive (SSDSA2SH064G1) The Intel Extreme SATA-II Solid-State Drive (SSD) offers outstanding performance and reliability for servers, storage, and high-end gaming systems. ☆☆☆☆☆	Intel	1 in stock	£599.99 (£521.73)	1	Buy ▶
Intel X25-M Mainstream 80GB 2.5" SATA-II Solid State Hard Drive (SSDSA2MH080G1) The Intel X25-M SATA-II Solid-State Drive brings a new level of performance and reliability to laptop and desktop PC storage. ☆☆☆☆☆	Intel	4 in stock	£262.99 (£228.69)	1	Buy ▶

通过替换标准的“库存有货”标签，Overclockers 在库存水平较低时显示准确的库存数目，这会促使想要购买的客户抓紧时间行动。因为错过这个机会，他们可能就买不着了。

7.11 退款保障，消除客户的后顾之忧

风险是完成访客转化的最大障碍之一。人们能相信你的公司吗？如果你的产品名不符实怎么办？如果客户发现了更好的商品怎么办？风险在客户脑中播下了疑虑的种子。因为这里有太多的不确定，所以他们决定不了是否购买你的产品或者进行注册。谢天谢地，我们有一个处理风险的好办法，消除客户对风险顾虑的最好办法就是退款保障。

退款保障意味着，在一定时间内，如果客户觉得不满意，可以把货物送回来然后取回货款。你不应当为这个感到害怕，如果你的产品是好的，很少会有客户进行退货。退款保障甚至对你的业务来说更有益，如果你不提供退款保障，而客户又非常希望把他的钱拿回来，他可以请求银行进行退款。这样你就会收到银行的借项冲回通知，你不得不在退款的基础上缴纳额外的费用。提供退款保障意味着不需要额外的结算费用，还可以获得更多的用户（低风险）对于公司的信赖。

7.12 鼓励用户试用

销售产品的一个最好办法就是让用户试用。用户试用时，他们就可以了解产品使用效果，这比仅仅看看产品或想一下要真实得多。他们在实际使用时融入其中。他们开始体验，了解产品如何工作并且如果是一款软件应用，他们会将自己的数据导入其中。总体来说，他们更多地参与了这个产品，购买的机会也显著地提升了。

如果你销售的是数字产品，如桌面或 Web 应用等，为什么不能让人们尝试一下？许多应用程序提供在线演示，人们可以登录和使用而无需购买。也有不少使用增值服务模式，即人们可以使用该产品免费的基本版本，而必须为额外的功能进行付费。这是个很好的主意，因为访客开始做两件事情：他们了解产品如何工作，他们保存了自己的数据。高级用户在你的产品上投入时间，因此在他们需要使用更高级的功能时倾向于使用你的产品，很可能会进行一次专业的升级。另一种让访客试用的方法是限制一个试用期。

Who Should Read This Book?



If you're working on the Web today, you need a copy of this book to understand exactly why some designs work, while others don't. It covers graphic design principles in a very straightforward and easy-to-follow way: you don't need to have had any formal art or design training to benefit from this book.

No-risk Money-back Guarantee

Of course we're *so confident* that you'll treasure this book for years to come that we're happy, as always, for you to try it risk-free for 30 days. If you purchase a copy of *The Principles of Beautiful Web Design* and you think it fails to be everything you wanted, we would like you to have your money back.

Simply contact us and we will see to it that you receive a prompt and courteous refund of the full purchase price minus shipping and handling.

What could be fairer than this?

Sitepoint 为其书籍提供 30 天退款期，也就是说，用户面临的风险是非常小的：如果他们觉得产品不好，就可以收到退款。



Pixlr，基于 Web 的图像编辑软件，为您提供了正确的演示链接登录页面。点击它可以让你即刻体验产品，而不必注册或泄漏任何个人信息。

7.13　别让邮寄费用吓跑了顾客

ForeSee Results[1] 的一项研究表明，通过对 30 家在线商店的 10500 项交易分析得出，34% 的客户因为在线商店不承担运费而取消了交易。人们希望看到

[1] http://www.foreseeresults.com/Thank_Top100_Mayo8.html。

购物篮中的价格就是最后需要支付的价格。在结算的最后环节，他们不想看到的是额外的成本。

提供免费的邮寄肯定可以刺激销售，但是却不针对所有网站都是可行的选择。如果确实需要收取邮费，那么一定要做到透明，提供清晰简单的邮寄选项。在这个阶段，你不想让潜在客户为他们需要支付多少邮寄费用而困惑，这应该是清晰明了的。最好的办法就是在客户的结算页，当客户选择邮寄方式后，更新显示总的邮寄费用，并最终体现在总价上。你还应当让邮寄选项清晰、易于选择，因为客户不想在支付环节上有所耽误。

7.14　消除选择麻痹

选择麻痹是营销领域的一个概念。描述了顾客面临太多相同类型、缺少区分的选择时的状态。例如，当你走进超市想要购买意大利面，可能会看到货架上有各种品牌、形状不同的意大利面。那么你会选择哪个？哪个更好一点？当选择太多时，你变得麻木而不知道选哪个好。选择麻痹不仅仅延长了顾客的购买时间，而且更容易造成他的懊恼。如果最后他选择了不喜欢的商品，他会后悔花了这么多时间、后悔没有选择其他的商品。为了帮助顾客克服选择麻痹，你应该让产品系列保持较小的规模，并且在产品之间有所区分。如果你仍然有很多的选择而无法再减少了，就给顾客推荐一款，例如最流行的产品或者服务计划。使用视觉效果来突出推荐的选择，让顾客一眼就能够认出来。不确定自己需要什么的人们可以很轻松、放心地选择推荐的产品，而那些高级顾客则可以很容易地挑选适合自己需要的产品。

30-day Free Trial on All Accounts

Pick a plan & sign up in 60 seconds. Over 8,000,000 contacts managed with Highrise.

Max	Premium	Plus	Basic	Solo
$149/month	$99/month	$49/month	$24/month	$29/month
TOP-OF-THE-LINE	FOR BIG GROUPS	OUR BEST VALUE	FOR SMALL TEAMS	FOR INDIVIDUALS
Unlimited users	Up to 40 users	Up to 15 users	Up to 6 users	Single user
50 GB storage	20 GB storage	10 GB storage	3 GB storage	3 GB storage
Unlimited deals	Unlimited deals	Unlimited deals	10 deals	Unlimited deals
50,000 contacts	30,000 contacts	20,000 contacts	5,000 contacts	20,000 contacts
Enhanced security	Enhanced security	Enhanced security	Standard security	Enhanced security
Sign Up	Sign Up	Sign Up	Sign Up	Sign Up
		30-day free trial on all accounts		

Max & Premium plans include Campfire Premium for free.

We also offer a free plan: 2 users, no files, 250 contacts.

Highrise 通过有效地使用视觉效果突出了为顾客推荐的计划。

Delivery conditions				
Type of delivery	Germany	EU	within Europa	outside Europa
UPS Standard	For orders made before 12.00 *) Delivery within 3-5 working days to centers of commerce 6,50 € (from 200,- € free of charge **)	For orders made before 12.00 *) Delivery within 3-5 working days to centers of commerce 30% Discount based on UPS price list	For orders made before 12.00 *) Economic delivery Optimum transit time ***) 30% Discount based on UPS price list	
UPS Expedited				For orders made before 12.00 *) Economic delivery Optimum transit time ***) 30% Discount based on UPS price list
UPS Express Saver	For orders made before 12.00 *) Delivery by next working day before 12.00 30% Discount based on UPS price list	For orders made before 12.00 *) Delivery by next working day 30% Discount based on UPS price list	For orders made before 12.00 *) Fast economical delivery. Please check exact transit time at www.ups.com 30% Discount based on UPS price list	For orders made before 12.00 *) Fast economical delivery. Please check exact transit time at www.ups.com 30% Discount based on UPS price list
UPS Express	For orders made before 12.00 *) Delivery by next working day before 10.30**) 30% Discount based on UPS price list	For orders made before 12.00 *) Delivery by next working day before 12.00 to centers of commerce 30% Discount based on UPS price list	For orders made before 12.00 *) Fast economic delivery. Please check exact transit time at www.ups.com 30% Discount based on UPS price list	For orders made before 12.00 *) Fast economic delivery. Please check exact transit time at www.ups.com 30% Discount based on UPS price list
UPS Express Plus	For orders made before 12.00 *) Delivery by next working day before 8.30 **)	For orders made before 12.00 *) Delivery by next working day before 9.00 to centers of commerce		

Badgepoint 的邮寄信息表。不要强迫顾客去分析这样的大表，而应该提供简单清楚的邮寄解决方案。

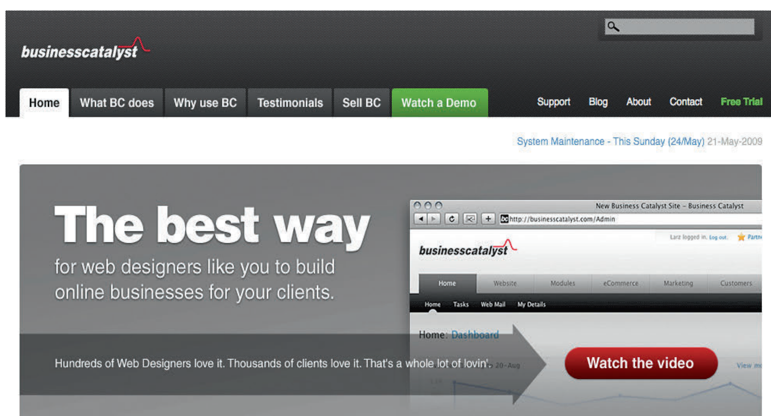
7.15 古腾堡法则

Gutenberg 图（或 Gutenberg 法则）是一幅描述“阅读重力”的地图。阅读重力是指在阅读时，我们的视线所试图遵循的路径。在西方，这个路径是从左到右、从上到下。

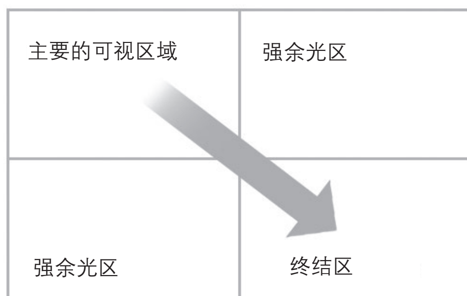
Gutenberg 图分为 4 个区域：

- 左上是主要的可视区域；
- 右上是强余光区；
- 左下是弱余光区；
- 右下是终结区。

Gutenberg 法则告诉我们，客户多数喜欢从左上开始看，最后终止在页面的右下。同时也告诉我们左下角得到的关注是最少的。



BusinessCatalyst 在登录页中使用 Gutenberg 法则布局文字介绍和希望用户点击的按钮“观看录像”按钮。一个箭头有助于引导访问者找到按钮。

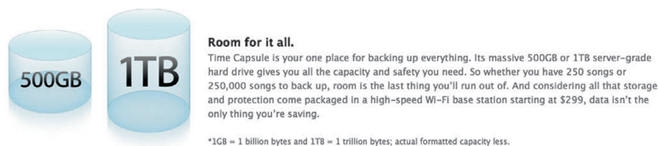


我们可以在一个普通访客可能很快浏览而过的网页中使用这个方法，通过将重要信息放在左上区域，一个需要访客点击的按钮放在右侧区域。注意，此法则在内容平衡分布的页面上表现最好。因此，如果页面中的特定元素有强的对比度和抢眼的颜色时，即使他们处于一个较弱的区域，仍然会吸引访客注意。

7.16　脚注

有时你需要详细的技术规格或详细的需求（例如，运行软件的最低要求）。在营销的版本中也许会使用同样的文字，但是可能会打破叙述的流程。这些细节通常不会促进产品的销售，因为它们是些简单的额外信息而且没有带给访客任何好处的内容。您可以通过将这些信息移动到脚注中来改善文字。

宣传文本将更薄，更着重于转换。同时，如果访客需要，他们仍可以通过阅读脚注来得到额外的信息。但是，请确保不要用脚注作恶。不要在正文中阐明了一些东西，而在脚注中违背它。例如，把实际不免费的东西说成免费。不是每个人都会读注脚，所以诚实地解释一切是最好的选择。记住，不要欺骗访客或给他们不正确的期望，那将最终导致访客的不满。



Apple 网站的脚注解释了存储的容量。

7.17　简化注册

注册表单是访客完成转化的障碍。因为它需要耗费时间和精力，几乎没有人喜欢它们。注册表单很可能是你转化访客的最后阶段，所以应该把表单做得短小、简单，确保在这个阶段损失尽可能少的访客。表单中不要问一些可选的信息，他们可以在稍后补充。保持表单的轻巧，减少访客填写的痛苦，加快访客填写的速度。将额外的导航和内容信息从页面中移走，只保留表单也非常有用。这样就没有其他的因素干扰访客填写表单。对于很多在线表单，访客经常在提交后才发觉自己所犯的错误。他们被导向到之前的页面，重新面对这些选项和列表。通过使用 AJAX 动态验证用户的输入可以节省用户的时间。将错误信息放在输入框的右边，一旦访客犯错，在提交之前他就可以进行修改。



Posterous 使用了特别的方式：你都不需要注册。只要发送一封邮件，就可以创建一个新的 Blog。

7.18　总结

我们已经了解有效销售的理论：推销解决方案而非产品；推销带给客户益处而非产品功能。我们也已经了解 AIDA 销售理论如何应用在网络中。我们还了解了促进访客转化率的实际技巧。

大多数的网站并不是艺术品，它们并不仅仅需要美丽得让人惊叹。更重要的，它们需要完成一项明确的任务，例如向访客销售产品、为读者展现博客。网站的设计应该为它的目标服务。当然，外观也很重要，它决定了访客对网站的第一印象，但是同样重要的是，不要为了设计而让网站走入歧途。

认真思考网站中每张照片的作用。考虑是否经过特定的处理能更好地帮助你的网站。网站设计不仅仅只针对功能，吸引眼球同样重要，但是只是美学上的好看并不能提升网站的表现力。因此，设计策略和决策都是由网站的目标和计划驱动的。

每个网站都是不同的，都服务于特定的客户群体，所以需要不断地微调网站的战略和战术。在一个网站上表现良好的手段在另外一个网站上不一定好。你如何确定哪个更适合自己？测试。目前有两种流行的测试方法可以帮助改进你的网站：A/B 测试（也成为分割测试）和 A/B 测试的高级扩展，多变量测试。

A/B 测试是在两个相对的页面上测试页面或者一个项目上两个不同变量表现

的过程。例如，如果你想试试两个触发行动的页面哪个更好。每个访客会随机访问其中一个方案，而且访客的转化结果被记录了下来。在收集了足够的数据之后，通过数据结果就能找到最好的方案。

多变量测试使用相同的原则，但是允许同时测试多个不同的变量。例如，你想测试网站 banner、宣传语或者市场广告的不同版本。多变量测试随机为访客提供一个测试项目，记录每种组合的表现。

尽管多变量测试听起来非常复杂，但是我们可以通过使用工具来简化这个过程。Google 的 Website Optimizer 是颇受欢迎的工具，可以进行 A/B 测试和多变量测试。更重要的是，Website Optimizer 是免费的，也就是说每个人都可以感受这两种优化方法的优势。

了解这些理论和实践方法只是开发和运营卓越网站的一个方面。另外一个部分是测试：发现哪些方法有效，哪些方法无效，并且进行相应的调整。网站不像报纸，一些更改是不可避免的。你的网站不会保持一个完成的状态，为提高访客的转化率，网站需要不断地改善和适应你的测试和优化。

Combinations　Page Sections

Analysis for: Aug 21 2006 - Aug 21 2006

View: Best 23 Combinations　Worst 23 Combinations　Download:　Print | Preview

Combination	Estimated Conversion Rate Range [?]	Chance to Beat Orig. [?]	Chance to Beat All [?]	Observed Improvement [?]	Conversions / Impressions [?]
Original	31.2% ± 3.0%	—	0.41%	—	125 / 401
Combination 11	38.9% ± 3.1%	99.0%	85.4%	24.9%	160 / 411
Combination 4	33.6% ± 3.0%	76.8%	4.12%	7.74%	133 / 396
Combination 23	33.4% ± 2.8%	75.9%	2.82%	7.17%	153 / 458
Combination 16	32.7% ± 2.9%	67.8%	1.56%	4.75%	144 / 441
Combination 10	32.6% ± 2.9%	67.5%	1.69%	4.67%	139 / 426
Combination 8	32.4% ± 2.9%	64.6%	1.30%	3.90%	137 / 423
Combination 22	32.0% ± 3.0%	60.0%	1.03%	2.69%	129 / 403
Combination 7	31.6% ± 2.8%	55.1%	0.45%	1.27%	143 / 453
Combination 14	31.4% ± 2.8%	52.2%	0.41%	0.57%	137 / 437
Combination 21	31.1% ± 2.9%	49.1%	0.40%	-0.20%	126 / 405
Combination 18	30.3% ± 2.8%	39.1%	0.11%	-2.79%	130 / 429

Google 的 Website Optimizer。

第 8 章

把网站变成非凡的品牌

Chris Spooner

“不同凡响”意味着值得关注。在网站开发领域，这意味着用户提起网站或者推荐给他的朋友。不断努力将你的站点打造成一个不同凡响的品牌，这样用户会自愿地信任或者偏爱你的网站，而这正是你建立和开发网站过程中的意外收获。

不管在什么行业或市场中，总是存在难以计数的相似的网站。众多的网站中却只有寥寥几个能够不断发展并获得成功。这些网站经常成为人们谈话的话题、作为获取信息的来源或者是一些流行产品的主页。考虑以上的情况，这些网站都可以归类为不同凡响，那么它们是怎么获得这样的成功的呢？

本章中汇集了大量对运营网站有用的建议和意见。其中很多的例子是基于一些高级的案例，这些技术可以用在不同水平的网站中。不管你是运行小小的个人博客还是大型的企业网站，每个建议背后的方法将帮助你把网站做到更好，做到更有名气；帮助你成为在行业或者市场同行眼中的权威；帮助你运营一个成功的社区，不断提高流量和销售额。最重要的是，让你的网站不同凡响。别忘了，仅仅是应用这些技巧中的一个就能显著地改善网站，如果同时使用所有这些技巧并且投入时间去做，你的梦想最终会成为现实。

举例来说，一个内容优秀但是外观平平的网站一定会成为不同凡响的品牌，同样一个外观优秀但是并不能提供顶级内容的网站也可以在一定程度上获得成功。本章中将会详细分析这些例子的缺点，并且逐步进行改进，最后提升这些网站的表现。

8.1 建立独一无二的设计

建立代表企业、产品或者服务核心价值和理念的视觉标识是创立任何品牌的第一步。通常都会从设计 Logo 开始，接下来准备宣传推广材料，当然这里

包括网站。要建立一个不同凡响的品牌，设计应当独一无二，能在众多同类网站中脱颖而出。同时要选择一些专业的、能够赢得信誉的高质量图片。独一无二的设计能够让人眼前一亮，而不会同平庸的设计混淆起来。那么，怎么样设计一个独一无二的网站呢？



Web Designer Wall 网站上线的时候，依靠吸引人的内容以及设计感，赢得众多访客的注意，现在这个网站仍然非常有魅力。

8.2　了不起的视觉设计

博客 Web Designer Wall 在网站的背景中使用了大量复杂的插画。在网站中使用大花设计，同时使用一系列精心设计的元素以及对细节的关注，使得网站不同凡响，而且很容被识别并被视为是创新的源泉。

8.2.1　注重普遍的设计原则

Black Estate Vineyard 网站使用了很少的颜色，也没有装饰的设计元素，但是

因为其注重结构、网格布局的设计以及完美的字体仍为它赢得了广泛的赞誉。回归基本的方法让它在众多有影响力和排名靠前的网站设计中脱颖而出，网站拥有了与主题相符的复杂、成熟的特性。



Black Estate Vineyard 没有过多的修饰，只使用设计的核心原则，便脱颖而出。

8.2.2 创造一些从来没有出现到过的东西

虽然 Silverback 网站页面中只包含了一个电子邮件的注册区域，它仍然非常引人注目。这是网站突出的视差效果造成的，这种技巧在其他的网站中还从来没有出现过。就是这一单独的特性吸引了大量访客的注意，它帮助 Silverback 在应用发布之前就赚了足够的声誉。

关键技巧

- 创建行业或者市场中独一无二的设计。
- 在所有在线设计中使用你选择的 Logo，传递同样的信息和样式。

- 注重细节，进行高质量的设计。
- 不要挤进用得滥俗的潮流或者趋势中。



Silverbak 网站背景中的插画效果吸引了很多人的注意，让网站看起来与众不同。

8.2.3　提供有趣又有用的内容

网站的设计给了用户第一印象，但是内容是在一个很长的时期内构建这个品牌。内容包括企业为了宣传产品或服务而在宣传页面中提供的资源，例如博客中的内容。值得信赖的高质量内容将最终为品牌赢得尊重，怎么做才能提供优秀的内容呢?

提供有用的信息

专业的博客连续不断地发布非常有用的信息，用户也能够从中学习和有所收益。这些内容紧接着就会被到处转发，并且在社交媒介中得到推荐。拥有了高质量的内容，网站作为专业信息来源就成为了一个不同凡响的品牌。

提供人们希望使用的

37Signals 是一系列必不可少的小型商业应用和合作软件的开发者。高质量的产品帮助网站树立了高水准的品牌。不管你的网站是宣传产品、公司或者服务，必须能够提供专业级的内容来逐渐制造话题和培养一批尊重你的内容的用户。

成为所在领域的专家

通过提供 Wine Library 方面的高质量的内容，Gary Vaynerchuk 已经成为该领域的专家。专业的内容不断增强作者的权威形象，反过来促进了公司或者网站成为一个不同凡响的品牌。

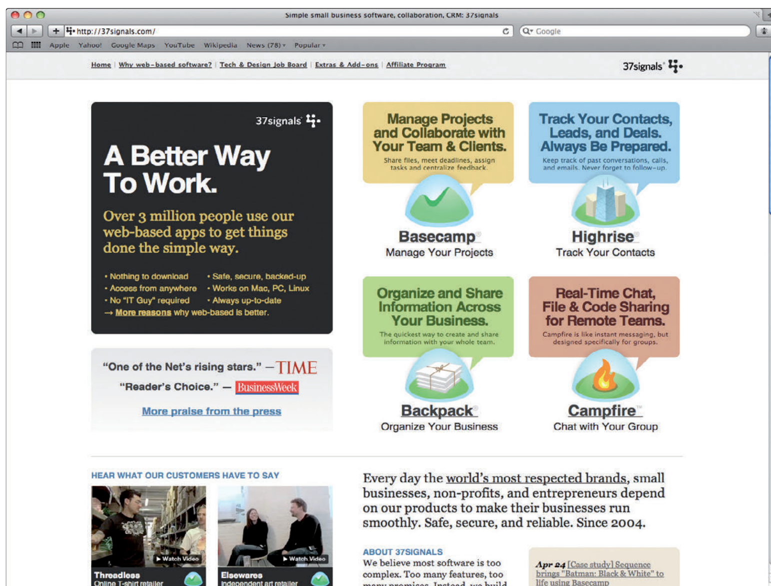


专业博客作者 Darren Rowse 通过不断地产生高度可用的内容，已经成为了一个著名的品牌。

关键提示

- 提供能帮助用户解决问题的原创内容，提供建议或教程。
- 向成功者学习并且不断深入：找到一个有用的观点，然后在此基础上不断深入。
- 不要以赚钱为首要目的，否则你网站的内容会缺乏质量保证。

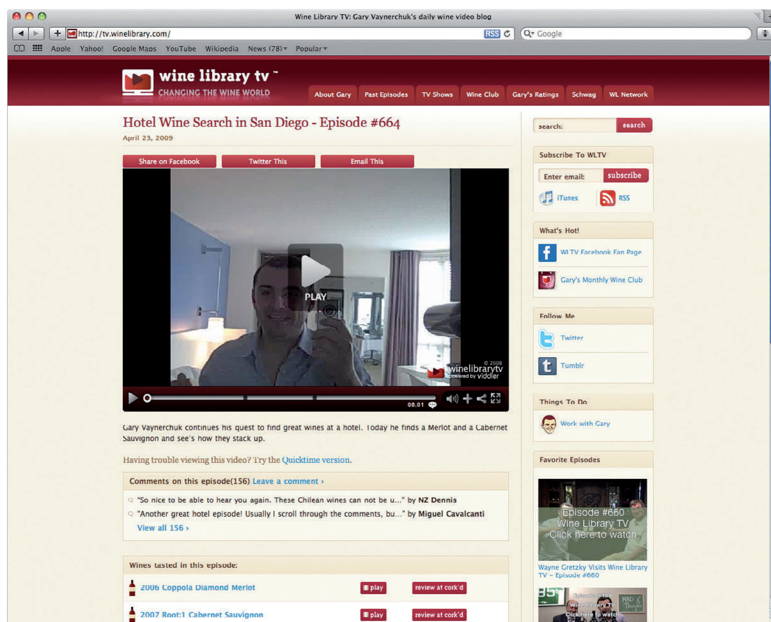
- 不要经常宣传自己。把精力放在帮助他人上，你会得到回报。
- 别放弃，生产高质量的内容需要时间和耐心，特别是刚开始的时候。



37Signals 提供了一系列优秀的应用程序，这正是其粉丝众多的原因。

8.2.4 解释新概念

令网站脱颖而出的一个最好的办法就是做一些完全与众不同的事情。那些使用新媒体来向用户传递信息的网站显然对用户产生了巨大的影响。视频内容成为一个可以将网站所有者从传统的、基于文本的网络互动中解脱出来的工具。那些应用视频作为信息或者娱乐目的网站很好地发展和扩展了他们的品牌。下面我们看一些例子，来了解新媒体是如何让网站受益的。



Wine Library 通过展示独一无二的视频内容向用户传达了其作为在线 wine library 领域权威的信息。

添加自己的理解

My Damn Channel 是一个允许喜剧演员、音乐家或者电影制作人创建系列视频的娱乐工作室。“You Suck at Photoshop”系列使用常见的屏幕截图式教程视频讲述了作者 Donnie Hoyle 十分有趣的生活。凭借独一无二的视频以及有用的提示和技巧，网站获得了巨大的成功，并且其他的视频也迅猛地传播开来，并且最终帮助 My Damn Channel 获得了 12 项网络大奖[1]。

[1] http://www.mydamnchannel.com/about.aspx。



My Damn Channel 发布了一个 20 集的风趣幽默的连载，介绍 Photoshop 的使用技巧。

紧追流行文化

The Onion Network 是一个基于当前热点话题即时发布虚拟新闻的频道。这些表面上看起来现实、严肃的视频，其内容其实充满了讽刺。这些创新的内容让网站作为娱乐之源越来越受欢迎，从而成为了一道亮丽的风景。

提供与流行话题相关的娱乐信息

Kevin Rose 和 Alex Albercht 合作举办的 Diggnation 口头讨论来自网上的新闻和最热的技术话题。这种轻松的表演方式提供了一种有趣的传递信息的渠道，并且口头的交谈和随机的话题也带来了娱乐价值。基于个人的理解让这个节目从其他众多的技术新闻网站中脱颖而出，让人觉得非常特别，并且得到了足够的曝光，有了众多的追随者。

关键提示

- 讨论网络上热门的概念，比如 TV 风格的系列叙事方式。
- 超越传统的博客内容格式来创建独特的内容：视频、音频以及更多选择。

- 增加个人的意见，同竞争者进行区分。
- 紧追流行文化来追随细分市场。



The Onion Network 不断发布现实播报风格的视频来评论流行文化和当前热点。

8.3　成为社区的一部分

任何品牌周围都会围绕一批人，他们形成了社区。这些人可能是公司、产品或者服务的忠实追随者，也可能是普通的用户。社区是所有品牌最有价值的财产。社区中人们可以分享他们的意见或赞美，可以自由传播你的品牌。当然，那些期望没有得到满足的用户也可能很容易地将公司、产品或者服务的负面消息说出去。在现实世界的社区中，利用互联网和社会媒体的资源，紧密关注品牌周边的社区动向是非常重要的。那么，品牌该如何应对社区的需求呢？

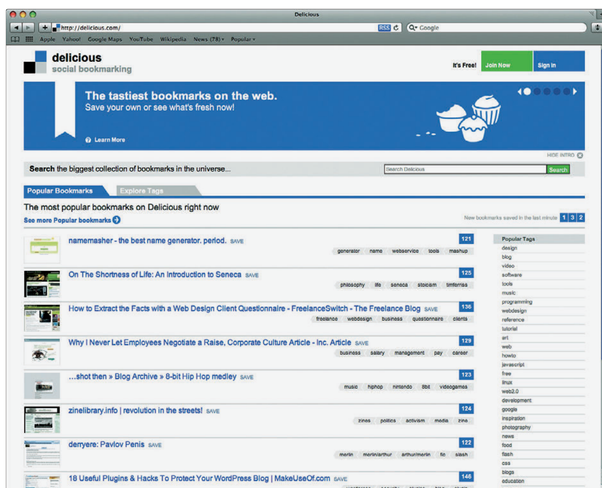
8.3.1　拥抱社会化媒体和人际互动

社会化标签网站 Delicious 监控着用户在 Twitter 上的关于其服务的问题、意见和评论，并且提供有用的帮助或建议。这种直接帮助用户的方式在公司和用户之

间建立了一对一的联系，让用户觉得他们的感受被公司理解，最后对公司有了很好的印象。



Diggnation 通过一种轻松的方式讨论从新闻网站 digg.com 获得的技术热点。



Delicious 使用 Twitter 来搜集谈论其服务的消息，并且马上就他们的问题和看法与用户取得直接的联系。

8.3.2 表现得友好和亲近

Headscape 是英国伦敦的一家设计公司。伴随着他们的设计项目，同时还开通了一个知名的网页设计博客和一个广受欢迎的交流博客，公司成员在上面对业界的众多问题提供建议和技巧（同时还有一些来自世界各地的贡献者）。让员工一露脸就能让用户更了解他们的性格特点，并且在公司和客户之间建立了一种友好的关系。在用户与公司员工接触的时候能有这样的优秀体验，他们自然会同公司的品牌联系起来。

8.3.3 做施与者，而不是掠夺者

Carsonified 的创始人 Ryan Carson，就是因为其收集的大量高质量的网页应用而成功。在一次采访中[1]，他归纳了创建一个成功网络应用的建议。其中一个就是让用户很容易支付下个月的费用，例如将账单分为 5 美元、10 美元或者 20 美元。同样，如果用户遇到问题，那么将最后一次交易的费用退还给他可以让用户立刻高兴起来。这种观点被认为是一个施与者，而不是只为了赚钱，这样能够培养用户对品牌的敬重和积极的态度。这种慷慨的态度留住了用户，并且良好的口碑会在用户之间口口相传。

关键提示

- 为你的公司、产品或者服务在各种各样的社交网站和博客建立账号，与顾客和用户进行直接的互动。
- 在这些渠道中发布实用的信息，鼓励用户参与并且成为网络中的一员。当然，别忘了答谢客户以建立一种双向的沟通。
- 让用户看到公司内部，而不仅仅是通过市场工具。
- 为用户提供有效、快速的帮助，维持用户对于品牌的积极态度。

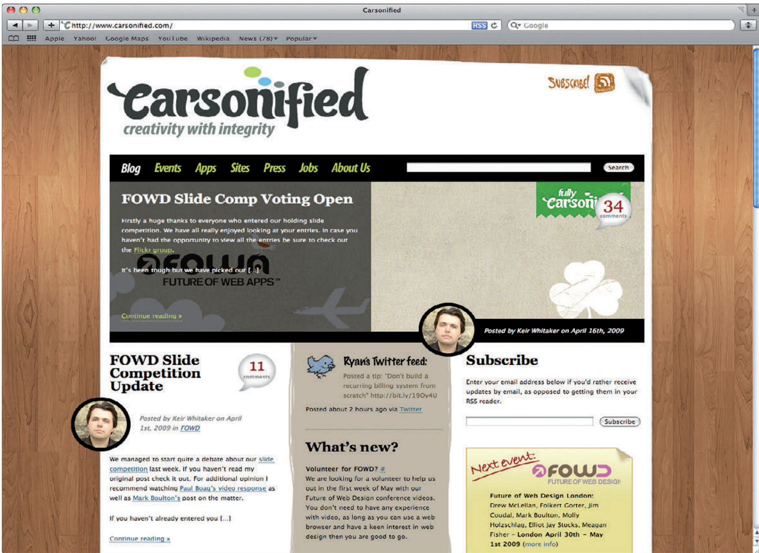
[1] http://boagworld.com/podcast/160-education-education-education。



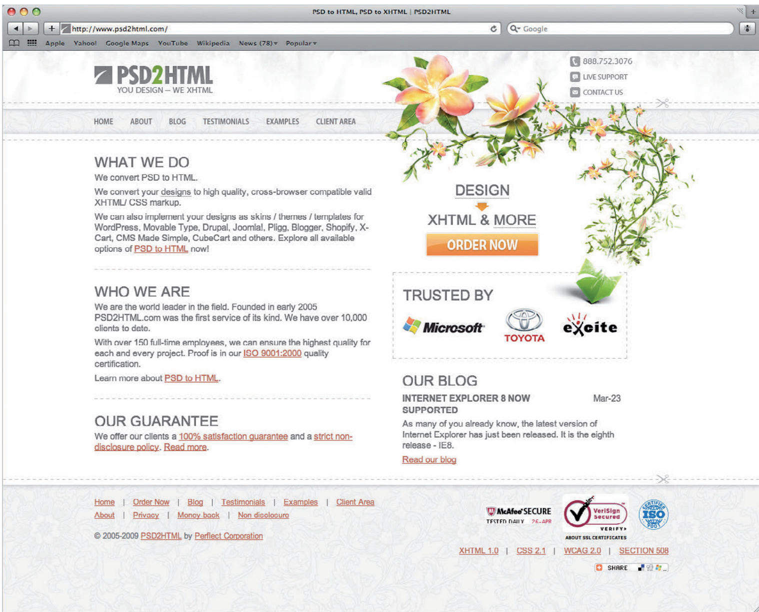
Headscape 支持用户通过博客、播客或者独立的 Twitter 账号与员工 Paul Boag 进行互动，这样可以留给用户平易近人的好印象。

8.4　进入人们的头脑中

建立品牌的一个亘古不变的目标就是让用户想到一个特定的主题时，头脑中第一时间就会想起这个品牌。要成为一个不同凡响的品牌，网站必须想方设法进入用户头脑中并且潜移默化地影响用户的选择。让品牌一出现就被接受是一个重要的步骤，那么如何做到呢？



Carsonified 认为给用户一些免费的体验是成为一个成功的网络应用的重要条件。



PSD2HTML 在业内其他的网站或者博客中做了广告宣传。

8.4.1 广告无所不在

PSD2HTML 公司懂得宣传品牌识别的好处。它的横幅广告出现在大量的广受欢迎或者高质量的博客中，这些都面向它的目标市场。这种大量出现的品牌名称和标识让用户在想寻找一家专长于 XHTML/CSS 转化的公司时首先想到 PSD2HTML。

8.4.2 成为话题人物

Clear Left 公司的 Andy Budd 经常出现在各种采访中，并且在众多的会议上做过网页设计用户体验的主题演讲。他还在传统出版物、会议、博客和 Twitter 上发起了一系列的讨论活动。这就让他和他的公司不断地在目标市场面前曝光，结果就是，他被认为是这个领域的专家。

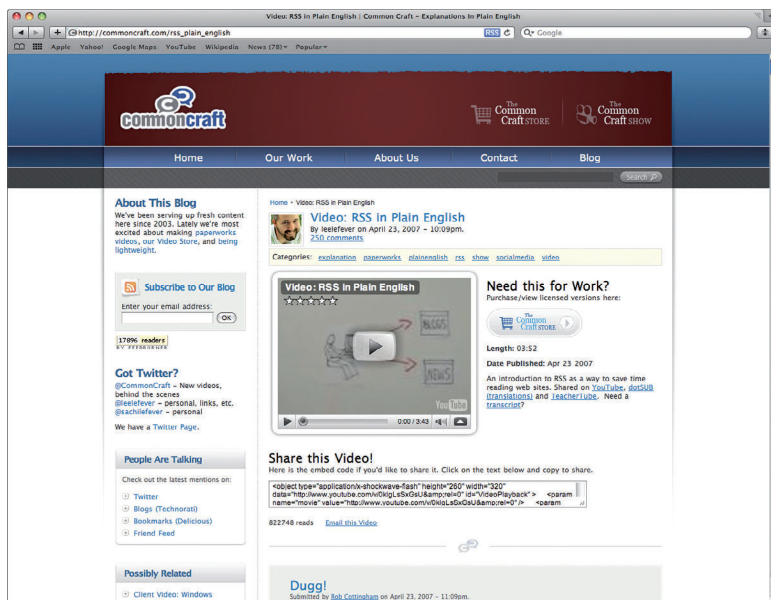
做一些与宣传公司看似无关的活动对建立一个良好的品牌形象有很大的帮助。帮助这个领域中有才华的人，支持一些教育项目，举办一些交流想法的会议都是很好的方法，这些无私的活动都有助于建立一个不同凡响的品牌。

8.4.3 创建"病毒"或者诱饵式内容

Common Craft 发布了他的第一个视频"RSS in Plain English"后获得大量的关注。一天之内，Digg 首页的视频就被浏览了 15000 次。这种类型的"病毒"或者诱饵式内容确实可以帮助创造大量的曝光，进而向一群新的用户介绍你的工作。最后，这些额外的关注极大地提高了 Common Craft 的品牌识别。

关键提示

- 在目标市场中宣传你的公司、产品或者服务。要尽力做到全面覆盖而不是提高点击率。
- 参加讨论、会谈或者一切能增加网站曝光度的活动。
- 创建那些可能在朋友和用户之间传播的有趣、有用的话题。
- 确保你的公司、产品或者服务成为众人讨论的话题。



Common Craft 的“The Rss in Plain Video”短片成为类“病毒”式内容，受到了来自四面八方的大量关注。

使用这些建议或技巧，毫无疑问可以将你的网站建设为受人尊敬的品牌。但是要记住，机遇稍纵即逝。这项工作需要日复一日的努力和耐心。对你的内容充满信心，帮助他人，传播你的看法，开始让你的网站变得不同凡响吧！

向专家学习——名家访谈与真知灼见 第9章

Steren Snell

许多成功的和知名的设计师愿意并渴望为那些希望提高的人提供指导。为了获得一些共性问题的答案，我们向一流的设计师和开发者提出了一系列的问题。参与者带来了大量不同的专业知识和技巧，所有这些有价值的见解都会帮助那些渴望提高的人。[1]

认识一下我们的专家组成员

Dan Rubin (webgraph.com)

Webgraph 创始人、主要负责人、Sidebar Creative 联合创始人

Jason Santa Maria (jasonsantamaria.com)

自由设计师、发言人

Paul Boag (boagworld.com)

Headscape 联合创始人、创意总监、Boagworld 创始人

Jeff Croft (blueflavor.com)

Blue Flavor 设计师

Andy Budd (clearleft.com)

联合创始人、用户体验总监

Collis Ta’eed (envato.com)

Envato 首席执行官（CEO）

Wolfgang Barthelme (bartelme.at)

Bartelme Design 创始人

[1] 本章的截屏是在 Flock 2.5 浏览器和 Windows Vista 系统上进行的。

Keith Robinson (blueflavor.com)

Blue Flavor 主要负责人、创意总监

Chris Spooner (spoongraphics.co.uk, line25.com)

自由设计师和博客作者

Jonathan Snook (snook.ca, sidebarcreative.com)

设计师、撰稿人、发言人

Tony Chester (onwired.com)

OnWired 创始人和运营总监

Elliot Jay Stocks (elliotjaystocks.com)

自由设计师、撰稿人、发言人

Khoi Vinh (subtraction.com)

New York Times 设计总监

Veerle Pieters (duoh.com)

图形设计师、博客作者，Duoh! 创始人

Chris Coyier (css-tricks.com)

Chatman Design 和 CSS Tricks 网页设计师、博客作者

Jay Hilgert (bittbox.com)

Bittbox 的设计师、博客作者

Dave Shea (mezzoblue.com)

Bright Creative、CSS Zen Garden（CSS 禅意花园）创始人、Mezzoblue 创始人、自由设计师

Darren Hoyt (darrenhoyt.com)

自由设计师

Henry Jones (webdesignledger.com)

Web Design Ledger 自由设计师和博客作者

Liam McKay (wefunction.com)

Function 设计师、项目经理

Nathan Smith (fellowshiptech.com)

Fellowship Technologies 用户体验工程师、SonSpring 创始人

Nick La (ndesign-studio.com, webdesignerwall.com)

N. Design Studio 创始人、博客作者

Larissa Meek (agencynet.com)

AgencyNet 助理创意总监

Jon Hicks (hicksdesign.co.uk)

Hicksdesign 设计师

9.1　设计和开发

9.1.1　关于网站设计开发最常见的曲解或误解是什么?

Dan Rubin："作为一种媒体，网站设计有些独特的东西"。"解决网站问题与以往其他事情相比更需要截然不同的思考方式"。以上根本不是事实，尤其对设计者而言。视觉设计的基本原理对那些需要在屏幕上展现的设计同样有效。图层交互设计、信息架构和其他特定媒体领域都是与媒体相关的专业，就如同建筑知识可能与环境图形设计相关，或者从材料的角度看，印刷和装订的知识与包装设计相关。一个设计者对此理解得越早，就越能熟练地把网站界面设计所需的多个层面的知识结合。

Jay Hilgert： 我最常遇到的误解是客户的认识。我遇到的 90% 的客户低估了制作一个网站的工作量。仅仅向设计师支付费用是不够的。整体的设计依赖于客户希望传达的内容和信息，大多数客户没有意识到他们为此要进行多少

思考。为客户主页设计的一段文字，你需要等上三个星期，这也许让人难以置信，但这却经常发生。

Wolfgang Bartelme：我猜通常大多数误解是一个设计，我指的是网站设计，能够一步到位。其实，这是一个渐进的过程。

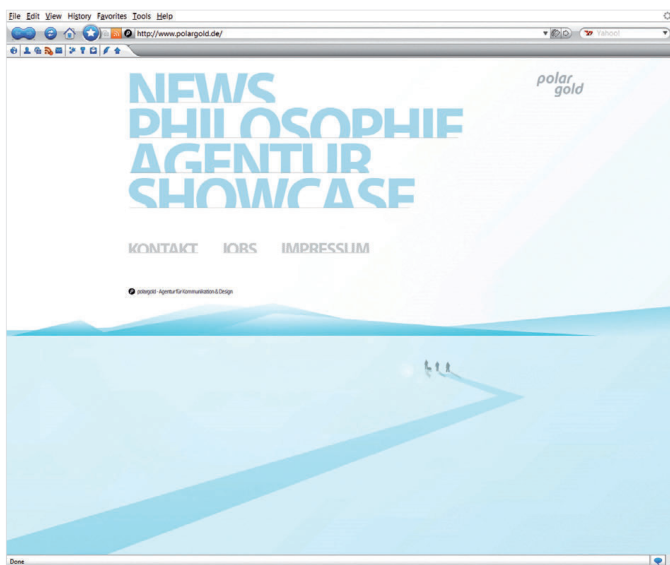
Keith Robinson：如果要获得成功，对工具（CSS 和 HTML 等）的精通是必须的。我觉得年轻的网站设计师（和开发者）的最大问题是狭隘地专注于工具，而不是去深入理解媒体。我认为知道如何写 HTML 和 CSS，以及对其他技术（Flash 和 JavaScript 等）有一个扎实的理解很重要，但是对与技术和工具无关的技巧的理解也同样重要。比如，知道 CSS 如何工作固然是好的，但是这不能代替对扎实的设计和可用性基础知识的理解。同样，掌握人际关系、时间管理等技能也是同样是重要的。

9.1.2　你们发现其他设计者常犯的可用性错误是什么？在设计网站时，可用性的哪些重要特点是必须考虑但却常被遗忘的？

Jonathan Snook：我发现设计者最常犯的可用性错误是出于美学的考虑使设计变得过于精致：链接或按钮变得不明显或者被隐藏起来，或元素靠得太近。用户需要知道他们点击的是什么以及他们为什么要点击它。否则，他们将忽略它。

Tony Chester：我发现设计者经常使用糟糕的导航拼凑网站。是的，网站可以是漂亮的，但是不能以牺牲内容为代价。是的，你可以使用内容管理系统，只要它不迫使你使用愚蠢的页面流。导航是至关重要的，尤其当网站的主要目的是卖东西、分享知识或收集用户信息时。如果网站不能轻易满足用户的目标，他们将会离开，而不管网站看上去是什么样。

除了这个，我发现太多设计者试图通过 Flash 和 JQuery 来使网站变酷，而这只会使网站变得难以使用。我讨厌因我的鼠标动作不够精确而立刻消失的动态菜单。因此要为人们提供较大的可点击区域，给他们的错误留一些余地。



一个由 Flash 构成的创意菜单。

9.1.3　你们怎么看 CSS 框架？你们是否认为设计师或开发者应该使用一个现成的框架，或者用他们自己的框架，或者完全不用任何框架？

Elliot Jay Stocks： 我不使用任何现成的 CSS 框架，因为我更倾向于用我自己的方式工作。所以，我有我自己的框架，它包含了 CSS、HTML 和一些 PHP 代码。我也使用我发布的免费的 WordPress 主题“Starkers”作为起点，所以我猜你可能会说某种意义上那也是一种框架。总体来说，我认为采用框架，尤其是你自己的框架，是个很好的主意，因为它们会大大加速开发过程。如果你要经常完成某种任务（实际上，这是很难避免的），使用一种框架会为你带来好处，因为不用自己重复开发。

Khoi Vinh： 坦白地说，我并不在乎。如果你是一个充满激情、雄心勃勃的设计者或开发者，对我而言，你是否使用框架一点也不重要，只要最终的产品与设计精神相符并能满足用户的目标和需求。只要你不用 Dreamweaver，对我而言就没什么问题。

Chris Coyier：如果你因为不理解如何实现你正在寻找的某种布局，而考虑使用一种 CSS 框架，那你最好还是不要使用 CSS 框架。CSS 只是一种简单的语言。重写它不会如此费力以至于你必须进行一些抽象才可以使用（比如 Ruby on Rails 中的 Rails 框架）。

CSS 框架当然有它们的用途，这个已经说过了。考虑一下一个充满网格、报纸类型的网站。每天都需要新的布局来使内容看上去更合理。维护这些网站的设计者需要的是不同的选择和不同的可能性，而不是限制。可以确定这些类型的网站在使用某种框架，使之成为可能。他们在使用 #home、.subColumn-3、div.first 和 .columnGroupB 这类该死的语义符号。

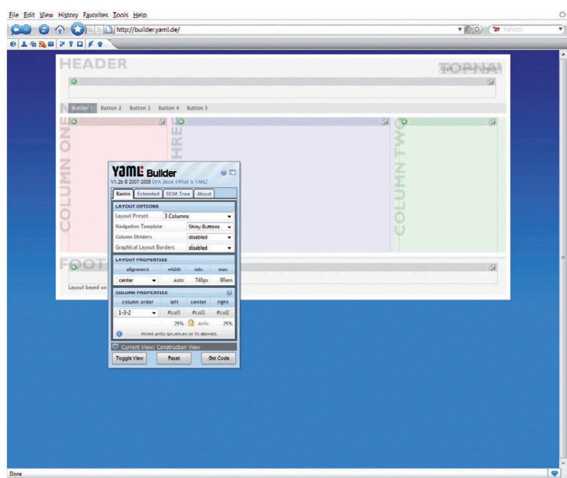
如果你处在这样情况下，创建你自己框架确实很有可能，但是在你开始投入工作之前，你应该好好了解一下那些流行的和久经考验的框架。对于入门者而言，我推荐 Blueprint, 960 和 YUI grids。

Dave Shea：我发现使用框架的两个主要好处。一是为网站编码时只需要较少的前端工作，二是较少的跨浏览器问题。对于前一点，多年来我自己开发了多种框架，它们在开始新项目时可以减少工作量。对于后一点，我已经太熟悉 IE 的怪癖，这些怪癖当时花费了我大量的调试时间。

对尝试某种第三方构架，比如 Blueprint、960 或 YUI，我一向不太在意。我之所以这样认为，是因为我自己为网站写了多年代码，为了达到布局的要求，我去研究某个框架所需的时间将等于或大于我自己编码需要的时间。这并不是说这些框架对其他人也没用。我确信它们将帮助一些人提高开发速度，并帮助初学者更容易地编写 CSS 布局代码。但是对我而言，框架不是问题。

9.1.4 你们认为设计者和开发者在搜索引擎优化中的职责是什么?

Darren Hoyt：如果你收取有竞争力的费用，你应该用符合逻辑和语义的 HTML 构建你的网站，这将是客户包含关键字的内容的补充。在理想状态下，你正在使用的 CMS 应该生成描述性的 URL。这些事情中的每一件都会吸引搜索引擎。这不应该是额外的一步。在我看来，这应该是标准。



在浏览器中工作的 YAML 在线网格生成器。

在开发者进行优化的“责任”范围内，所作的一切都取决于合同中的规定。在网站上线前，我会对客户解释，在编写优秀的代码之外，我不能确保获得绝佳的搜索引擎排名。我鼓励他们仔细思考他们上传的内容和他们在与哪些人交换链接，因为在 Google 看来，这是唯一能够为他们带来诚实的相关度的东西。

从个人的角度，在用户面前，我从来不把自己定义成 SEO 专家。我更倾向于把他们描绘成在提高排名方面具有经验的人。但我不是指那些靠通过不道德的方法欺骗 Google 或依靠无效或过时的方法的人。

“开发者在 SEO 中的角色至关重要。”

Tony Chester： 开发者在 SEO 中的角色至关重要。清晰的、符合标准的代码将会为排名带来奇迹。当 Google 的机器人到达你的网站的时候，它不想在无用的表格、过时的字体标记和不连贯的代码间浪费时间。它只要一件东西——内容。这也是开发者的切入点。为机器人提供它迫切需要的内容是关键。有一些人会做清晰的标记和创建分层次的内容。有的人会为你创建 SEO 友好的 URL，并创建 Google 深爱的 XML 格式的站点地图。没有合理构建的网站，SEO 将是一种挑战。

Henry Jones：SEO 可以分为两类——站外的和站内的。站外的 SEO 基本上是指指向本网站的链接数量。开发者对这一点没有太大的控制力。但是站内的 SEO 包括使用正确的 HTML 标签、创建坚固的网站结构、良好地使用内部链接，以及对搜索引擎一般工作原理的理解，开发者应该对所有的这些事情负责。

9.2 设计过程

9.2.1 当为某个客户开发一个项目时，你们第一步将做什么?

Darren Hoyt： 在项目启动会议上，我会尝试通过解释为什么要把访客放在第一位考虑来确定基调。这很快会减少客户的古怪念头，以防止个人的审美癖好破坏设计过程，这也会使客户确信我作为一个设计师，不是在为自己的利益设计一个作品。我们双方需要协同工作，来为访客创造最佳的体验。

如果客户还没有考虑过访客，我们会预算出一些时间，研究行业竞争、做标注和模型、更加清晰地描绘设计所面对的访客的特征。

在开启动会议时，其他一些需要获得共识的事情。

1. 理想状态下，客户应该至多只提供一个联系人。可以理解，这个人会从合作者和他们的上级那里收集反馈，但是不要允许更多的人通过电话或邮件进行反馈并给出相互矛盾的意见。
2. 如果像 Basecamp 这样的系统被用于追踪里程碑事件，有必要对项目计划做一个整体回顾。即使时间发生变化，对客户而言它也是一个可视化的提醒标识，提醒我们将会按一个常规的计划一同工作。
3. 在这一点上，如果设计工作期间中有任何假期或空档，应该及早提出来。我提醒客户，搭建一个网站是双向的，需要双方持续投入。有时候，这会把客户吓跑，但是直接了解客户是否会履行他们在交易中应该承担的义务是有好处的。
4. 为了进行预算，应该确定一个版面设计的过程。有些客户想要看到 3 个不同版本的主页，而有一些会要求唯一的主页和多个不同的内部页面。一些项目依赖于一个预估价格，一些则是固定价格。这取决于项目，没有什么可以大惊小怪的。

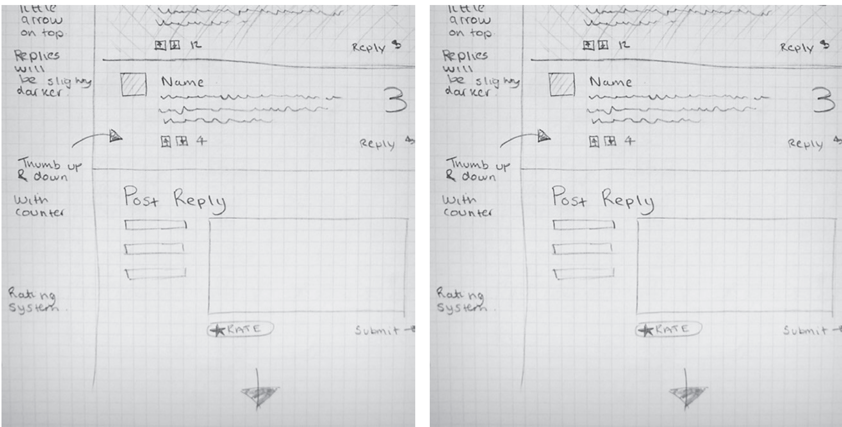
5. 总之，内容材料应该在版面设计之前提交。提醒客户，如果设计师在没有得到正确信息之前就开始版面设计，这个令他们花费不菲的网站将会大大受损。很快设计将仅仅变成装饰。

Jason Santa Maria: 我总是从跟客户谈他们想要达到的目标开始，如果可以，我会和他们的网站访客或用户交谈。每个项目开始时都要进行调查和探索，所以我能在信息充足的情况下做决定。

Chris Coyier： 确定目标是理想的开始方式。跳过这一步，直接开始设计是如此具有诱惑力。你可以跨越式地搭建一个漂亮的网站，但是美观对销售产品、使访客获得资讯、构建社区或达成其他客户想要真正达成的目标，并不一定是必需的。

挑战在于客户也许不能清楚地表达他们的目标。作为设计师，作为在建设网站方面富有经验的人，我们需要帮助他们描绘出那些目标。如果你做了这一点，并且完成和超越了那些目标，你的客户将会乐不可支。

9.2.2　在你们把草图或原型转化成一个实际的网站时，遇到的主要挑战是什么，你们如何克服它们?



经过重新设计的 Smashing Magazine 版式。

Liam McKay： 对我而言，最大的挑战是限制我所做的事情并专注于那些必须的事情。我并不会因为缺乏创意而纠结。正相反，在构建原型的过程中，我

的思想会走得太远，我想到了太多该包含进来的东西，这使设计面临不堪重负的危险。令我较难处理的是，分辨出哪些想法是实际需要的，哪些只不过是头脑过于兴奋的产物。我做的项目越多，在构思过程中发现哪些是实际需求的好处就越明显，它可以帮助你专注，并使你在那些可能并非必需的事情面前会收敛一些。这并不是说，你不能尝试那些标新立异的东西，它更多是一个关于通过基本要素构建一个坚实的基础和在此之上进行建设的案例。

Jason Santa Maria：一直最耗费我精力的事情是，寻找把草稿中自然流畅的本质带到完美的数字世界的方法。我发现处理这个问题的最好方法是，尽可能远离电脑。我一步步勾勒并细化出我的构思，把过程分为若干个循序渐进的阶段，每一步都添加新的内容，比如，色彩和更加优雅的版式。

Nathan Smith：一个项目中最具挑战的部分是信息构架。人们很容易跳过这一步直接开始视觉设计或开始编码，但是如果你对你正在构建的东西没有一个清晰的认识，你的努力将变成徒劳。为了降低不确定性的影响，我尝试通过高级的线条模型进行网站布局。我喜欢使用 OmniGraffle 进行建模。对于大一点的项目，我将使用分类学或思维导图对网站所要包含的各种信息进行分类。对于思维导图，我喜欢使用一个叫 FreeMind 的免费软件。它使我能够快速地把不同的节点和相关的主题连结起来。在花了一些时间把元素摆放完毕后，我通常会对信息架构（IA）有了一个充分的了解。

一旦我知道我要构建的东西，并确定了一个基本的页面布局后，我将投入到实际的界面设计中去。对于这一点，我发现没有什么软件可以和 Adobe Firework 媲美。它显然是我的武器库中用途最广泛的视觉设计工具。虽然其他一些人可能更倾向于使用 Photoshop，但我确信 Firework 更适合网站设计。

我一般喜欢在网格上进行设计。通常，我将会在 12 或 16 列的网格上开始设计，看一下什么比例最适合我将展现的内容。这并不是说所有的设计都需要使用 960 像素的网格或完全不使用任何网格。它只不过是给我一个起点，在一定限制条件下开始设计比面对一张令人发慌的白板开始设计要容易些。

我遇到的最常见的挑战是如何使一个网站跨浏览器兼容（尤其使它在 Internet Explorer 6 中工作）。

Chris Spooner：我发现在把草图转化为可视化网站的过程中，主要的挑战是

精确地创建并部署元素。粗略的草图把整个布局较快地整合到一起，但是线条和页面元素并不成比例，因此需要准确细致地重新制作。

我用来克服这个困难的主要方法是在网格上进行设计，同时也会借助应用软件提供的一系列页边空白和标尺。

Nick La：为了解决这个问题，我避免使用 IE6 不支持的 CSS 属性，或直接使用 JavaScript。

Jay Hilgert：当设计一个网站时，即使在最早的阶段（在 Photoshop 中），我最专注的事情就是功能性，特别是用户界面。我会问自己一系列问题，比如，“在需要为用户提供的信息中，什么是最重要的？”。通过这种方式，我不会偏离方向，而直接去考虑美学问题，或者我所谓的“脸蛋儿”问题。

我首先会按内容的重要性建立一个大纲，这将帮助我决定如何以及在哪儿创建导航（我的目标是让任何浏览网页的人能够在页面载入后不到一秒的时间内，清楚地知道这个网站是干什么的）。

这项工作一旦完成，我开始处理外观和感受，尽我所能使它们与内容、目标和网站的访客契合。在这个阶段的挑战是使它看上去美观、同时尽可能地减少载入时间、优化图片、检验标记的有效性以及编写硬如磐石的 CSS。测试和调试是最后的挑战，通常 IE 是最糟糕的浏览器。

“测试和调试是最后的挑战，通常 IE 是最糟糕的浏览器。”

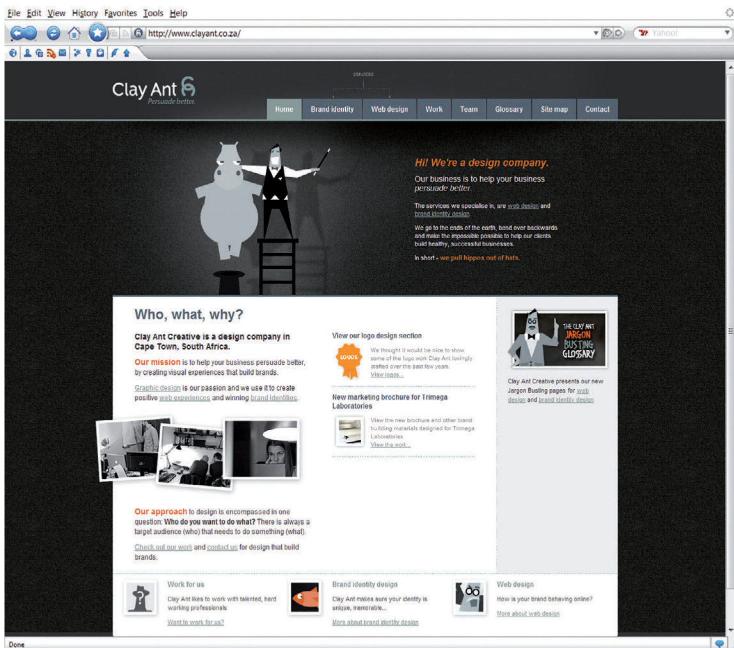
我该如何战胜这些挑战呢？大量的咖啡和震耳欲聋的音乐。

9.2.3 “如何确保一个网站能够真正抓住某个企业的精髓?

Jonathan Snook：这是很困难的，因为这也关系到一个企业希望怎样展现自己的精髓。通常，企业希望自己像竞争对手那样。理解一个公司是重要的，它的核心价值是什么？它在努力吸引的市场是什么？它现在的市场是什么？接受这些信息，把它和任何企业可能已经建立好的品牌结合起来，并努力把这些信息结合。记住，这不是一项关于投影效果和最新技术噱头的工作。这项工作是创建一些抽象的东西，这些东西会引导访客去完成任何他们已经开始着手去做的任务。

Larissa Meek：设计前要明确目标，比如：

- 网站要为客户做些什么？
- 网站要为访客做些什么？
- 客户最希望推广的东西是什么？
- 访客最可能做的事是什么？



一个商业网站示例。

如果能同时把握企业目标和用户目标，那么其他任何事情都会落到实处。设计最难的部分是它的主观性，每个人喜欢的东西都是不同的。所以，创造一些你认为是正确的东西，并且能够通过合理的理由支持你的设计决策。如果你知道目标，并且为这些目标工作，你就会抓住企业的精髓。

Nathan Smith：在我开始为客户设计任何东西之前，我总是努力理解两个重要的因素。首先，网站的用户是谁？其次，客户希望传达的理念是什么？我

坚信，通过首先考虑目标用户是谁，设计师可以定制出方案，甚至在他们制定方案时能帮助客户传达出他们的“声音”。

要传达什么样的理念，这主要依赖于什么使客户与众不同。

当为非盈利组织设计网站时，理解该组织的目的和目标非常重要。你很有可能需要对那些正在接受他们帮助的人进行案例研究。

当为企业服务时，最好知道它的竞争对手是谁，以及他们的核心产品和服务是什么，这样做的目的是突出客户的特别之处。

Dave Shea： 理解企业。我最成功的一些作品是建立在与客户公司中的重要人物谈话并努力理解公司的愿景和服务对象的前提之上。一旦准确地理解了客户，事情将变得更加容易，就是接受或拒绝那些我认为将会影响理解的设计理念。

Paul Boag： 在 Headscape，我们有一整套的方法专门用来捕捉企业的精髓。这个过程有时包括情绪收集板（mood board ）和参考网站（reference website）。另一些时候，它包含更多的抽象技术。我最喜欢的是“设计目标”（design objectives）和“网站角色”（website personas）。

设计目标和企业目标类似。对于企业目标，你通常会为网站需要完成的任务创建一个项目列表。对于设计目标，你的列表会包括你想要传达的信息和情感。

网站角色和用户角色具有许多共同之处。用户角色本质上是理论用户的虚构传记，包括他们的生活方式、工作、教育和其他人口统计信息等方面的细节。网站角色与之类似，但是会问这样的问题，“如果你的网站是一个人，那他会是什么样的人？”你的网站是一个时髦的少年还是一位中年商人？这些角色有助于定义一个网站的本质，并将它表达出来。

9.2.4 对于某个特定项目，你如何确定哪一个内容管理系统（CMS）更适用？

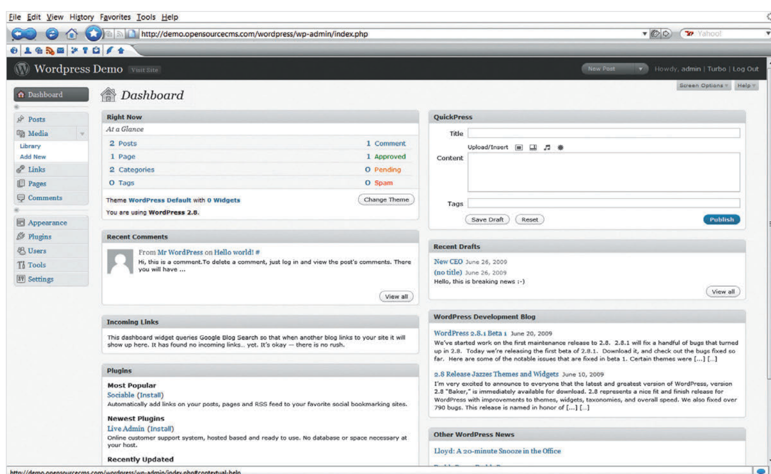
Darren Hoyt： 从项目启动会议开始，与客户紧密合作，并对他们的需求进行认真核查，这会使决定用什么 CMS 可以让事情变得简单一些。我通常在

这些会议的前一部分帮助客户专注于诸如读者和主题之类的大问题。

在会议的后半部分，我们会一同工作，来确定那些支持主题所需的一系列特征。头脑中带着松散的网站结构，我们开始对需要静态页面和需要定制数据类型（比如日历、新闻档案和房地产列表）的部分进行区分。

当会议结束时，我们已经有了一个可以进一步思考的“解决方案”。对于中小型企业，WordPress 附加一些插件就可以胜任。尤其近些年以来，文档数量大大增加，插件的质量和数量也有所提高，客户开始喜欢控制面板。一些开源软件，无需费时就可以证明它们的持久威力。目前，WordPress 具有主流的影响力，以至于许多客户实际上是慕名而来。

“一些开源软件，无需费时就可以证明它们的持久威力。”



WordPress 引擎的内部。

如果客户对数据的输入和输出要求非常特别，而且不想碰任何代码，ExpressionEngine 和 Django 也是不错的解决方案，它们都有顶级的开发者支持。与 WordPress 相比，掌握它们可能会有一些难度，但是它们的控制级别会更高一些。开发者可以按客户的需求定制管理环境，如果客户经常需要进行维护，这将具有很高的价值。有一个灵活的控制面板是至关重要的，它可以应付接下来的一两年内客户想要进行的重大改变。

我还认为对于开发者而言，偶尔留意一下 CMS 讨论社区的动向也是一种很好的方式。顶级的开发者是否支持它？论坛是否活跃？文档是否一直在更新？插件和扩展是否有用？是否规律性地进行着升级？创始人是否通过提供状态报告和未来的开发计划来吸引公众？公众是否对这个 CMS 的兴衰感兴趣？很显然，没有人能够预见未来，但是如果一个 CMS 很快淡出人们的视线、支持和文档越来越少，你最好要研究一下，并确保你的客户选择另外一个更具前瞻性的解决方案。如同理财顾问的职责一样（监控股票），网站开发者应该清楚他们推荐的软件的第一手信息。

Paul Boag： 在为客户选 CMS 时我会参考两件事情，即功能和支持。大多数客户需要相同的核心功能（页面生成和管理）。然而，除此之外可能会有很大的不同。你需要考察搜索、授权、工作流、评估管理和定制化等不同领域。我也花了大量的精力关注 CMS 中的编辑器。客户将花费大部分时间与之打交道的东西，它们中的很多却非常糟糕！

你需要考察的第二件事情是支持。你需要确定你选的 CMS 将会被长期支持，这样用户才可以得到他们需要的帮助。这里需要考虑的是安全（是否经常发布补丁）、开发团队、文档、培训和交流社区。

Chris Coyier： 这个问题的理想答案是，我首先评估一个项目的准确需求，然后根据这些需求对每个主流 CMS 的特长进行测试，并作出相应的选择。当然，事情从来不会这么简单。有时在事情进展一段时间之前，很难知道一个项目的准确需求。有时客户对要用什么 CMS 已经有了一些想法。有时遗留的问题会影响你的选择。

但是最重要的是，选择一个 CMS 和选择一辆新车不一样。对于一辆汽车，即使你从来没有开过它，你也知道该如何开它。插入钥匙、启动引擎、挂挡、踩油门。不幸的是，对于 CMS，事情可能会复杂很多，没有什么通用的准则。所以你不可能一下就找到“完美”的 CMS。学习如何使用一个 CMS 后端、设计前端并定制开发，需要经历一段不短的学习过程。

我实际上首先会选择 WordPress。WordPress 是我所知的最好的 CMS，它在我为客户开发的 80% 的网站上性能良好。如果我可以用 WordPress 创建一个网站，我就会用，否则我将寻找其他的替代品。需要说明的是，这和懒惰无关。

我曾经用 Drupal、ExpressionEngine、Sweetcron、Pinncale Cart、Magento 和 Cube Cart 搭建过网站，我自己也写了一些 CMS。选择你最了解的东西才会省时省力。

9.2.5 你们如何选择配色方案?

Collis Ta’eed：当为一个设计选择配色方案时，我最先需要考虑的是 Logo 和品牌。我会浏览我必须在网站上使用的那些特定的图片和其他的素材，因为如果要突出它们的话，这对色彩进行匹配很有意义。如果外部的东西对决定配色方案没有什么帮助，我经常会转向 COLORlovers 以获得调色灵感，因为它能使我远离老套的配色方案。

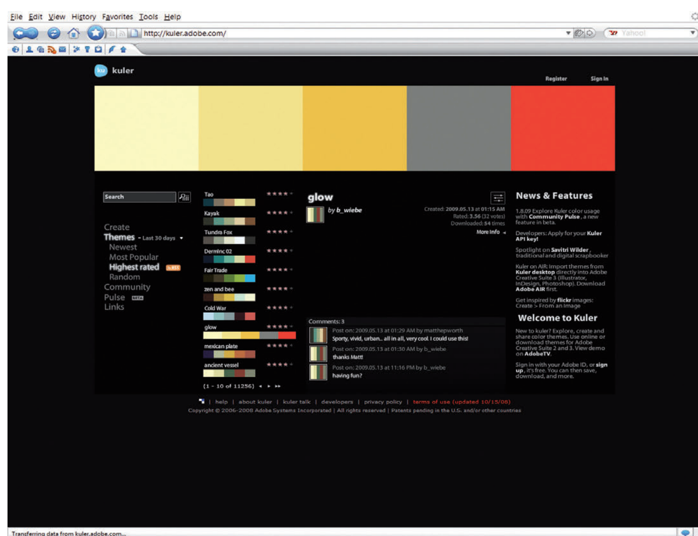
Larissa Meek：通常，我会基于客户的品牌指南确定配色方案。然而，如果开始一个新的项目，我会从图书、杂志、插画和现有的网站中寻找灵感。有时候这是一个有机的过程，合适的色彩通过设计探索被发现，因为结构和可用性在交互设计中占有很大的比重。

我从主页开始，选择几个主色和配色，它们会产生对比，这是创建一个高易用性网站的一种方式。一旦整体外观和印象获得认同，我会把这个配色方案用在网站的其他部分。

Elliot Jay Stocks：我不确认我在选择配色方案上有什么特殊的倾向。它取决于是否必须与品牌紧密结合，客户是否已经为某一种颜色确定了一种基调，以及是否必须激发某种特定的氛围。如果我没有思路，我会到处看看或浏览像 Adobe 提供的 Kuler 这样的配色网站。

Veerle Pieters: 首先，我会看一下公司是否有色彩样式指导。如果没有，在进行网站设计时我会从三个颜色开始：一个深色和一个浅色，以及一个和前两个颜色对比明显的亮色。这个技巧可以用来保持和谐。然而，很多的配色方案是通过直觉选取的，它们可能来自像 Adobe Kuler 这样的在线资源或任意一张图片。在过去做内景设计时，我也使用来自颜料样本中的色彩。这没什么秘密，任何时候我都可以分享这些成果，因为有时候我似乎在色卡中发现了一个不错的配色方案，但是当把它应用在大一点的画布上的时候，效果却不怎么样。实验是这个过程中的一个重要环节。

Jon Hicks：作为色盲，我倾向于依赖那些通过检验和测试证明能用的配色方案，或者从我的周围选择配色方案，比如海报、活页、照片以及大自然。每一个方案都有一种感觉和含义。哪个正确，这取决于未来的用户。比如，对于 ReallyWorried.com 这个品牌，它使用了宁静的、女性偏爱的色彩（粉红色和紫色），而刺激性的颜色将产生完全错误的感觉。



Adobe 的 Kuler 需要 Flash。

9.2.6　你们在每个项目中字体版式的通用原则是什么？

Jason Santa Maria: 我只是尽力专注于容易被忽略的小事，如合适的字体大小、和谐的字体组合以及正确的标点。

Khoi Vinh：就是使文字看上去不错并且可以阅读。

Jeff Croft：每个项目都不同，很难说我为每个项目都做了什么相同的事情，但是以下几点确实可以应用到我所做的 95% 的工作中。

- 设定网格基线。
- 选择一个字号，并一直使用它。
- 只限制自己使用三种及三种以内的（此种情况很少发生）字体。

- 设定合适的行距（leading length）和长度单位（measure length）。
- 使用合适的符号，比如正确的引号和撇号。
- 在保证效率的同时要尽可能精美。

9.2.7 你们怎样才知道一个设计做完了？

Wolfgang Bartelme：在理想状态下，一个设计在增加或减少元素就会降低可用性或吸引力的时候就算完成了。然而，在现实生活中，时间和（或）预算的制约往往在设计完成后才会清楚。即使你宣称一个设计“即将”完成，新的要求很快就会到来，你很快就得调整甚至重新思考你的设计。

Collis Ta’eed：很难知道该在什么时候停止设计。理想状态下，我会在第一次感觉快要完成时，尽量休息一两天。离开设计一段时间有助于头脑清醒，以便让我可以更加客观地看待它。时常，当我回来重看这个设计时，我会发现很多的瑕疵，于是开始通过微调进行修整，对细节进行润色，甚至有时会重新设计。当我发现任何改变都会使设计变得更糟而不是更好时，设计就算完成了。

Jon Hicks：在做设计时，你总会到达一点，你感觉只不过是在为它添加一些无关紧要的东西。在做一个项目时，我会把它放下一两天，然后再回到项目中，移除额外的修饰，把它精简到只剩下必需的元素。那时，我知道设计已经完成了。它有其需要传达的每一件东西，除此之外，没有别的东西。然而，对我而言，一个设计永远不会完结。我总是会回头看这些项目，看看哪里还可以完善，“只要我愿意……”。

Nick La：对我而言，设计从来不会结束。即使网站发布后，我也经常可以根据反馈对其进行改善。

Khoi Vinh：这是个很大的问题。很难知道什么时候才算完成。当然，你在追寻一种一切都处于平衡、和谐的状态，每一个元素都各居其位，按逻辑的、连贯的、还有一点振奋人心的方式展现自我。它也有另一种含义，就是一个网站的设计方案永远不会完结。就像一张照片，在某一时刻抓拍，照片中的大部分元素井然有序，但是你所看到的仅是某一时刻的景象，一个状态之后还会有另一个状态。

很难说出网站什么时候可以做好进入公众视野的准备，我其实不想过分思考这个问题。

Keith Robinson：这难道不是一个充满陷阱的问题吗？对于网站而言，设计实际上永远都不会结束。很难说出网站什么时候可以做好进入公众视野的准备，我其实不想过分思考这个问题。遵循你的直觉，在你觉得合适的时候就发布网站，只要你活着就不要怕做小的改动。



密切注意这个展示网站的版式。

9.3　自我提高和技巧积累

9.3.1　你们是否认为设计者和开发者应该接受正规的教育和培训，而不是靠自学?

Veerle Pieters：这实际上取决于个人，但是在大多数情况下，我会推荐通过接受正规教育以获得更好的理解。就我个人而言，在网站方面我没有接受过任何正规的教育，因为在过去它根本不存在。我主要靠自学，但是在学校里学到的设计基础知识使我受益匪浅。我认为关于设计和开发集中的教育会产生很大的不同，但是持续不断的教育才最有意义。持续教育、正规教育之后的培训同样重要。因为我们在为互联网、潮流和技术变革进行设计。离开学

校后你的学习并未完结。在你的职业生涯中，它是一个持续的过程。

Tony Chester： 这因人而异。教育的好处在哪里？我曾经跟一些出自知名学府的开发者一同工作。

他们具有关于最佳实践的基础知识，他们可以把它应用到任何他们正要使用的技术中去。在这些案例中，教育是有帮助的。在另一方面，我收到过一些刚毕业的计算机专业研究生的简历，他们除了 Java 之外什么都不知道。他们也具有同样的知识，但是他们似乎无法把它应用到实际工作，对这些人而言，正规教育并没有什么优势作用。

同样的事情也在设计师身上发生。如果他们能把色彩学、版式和布局的知识应用到网站上——尤其是如果他们学的是平面设计——他们就做得很不错。我遇到过太多的设计者，他们在 Photoshop 中能做出很漂亮的东西，但是对导航、可用性和其他对创建优秀网站至关重要的东西一无所知。

同时，业内很多人完全是自学成才，他们业绩非凡。对我们中的很多人来说，当我们在学校读书时，网站还在萌芽期或者根本不存在，所以我们只能在它出现后进行自学。最后，我认为最好的选择是，有一个较强的专业背景，但同时具有不断学习的动力，以及能够不断扩充对新媒体的认识。那些不被所用工具限制的人，将会是最成功的人。

“我们需要适应和成长。毕业并不意味着万事大吉；学习是一个贯穿你的职业生涯的持续过程。”

Andy Budd： 有这么多的人自学成才真是不错，因为从这一点可以看出，人们对这一行业具有多么大的动力和激情。然而，令人难以置信的是，他们可能缺乏一些从事这行所需的基础知识。对于网站设计者而言，这些缺失的基础知识可能是色彩理论、网格设计和版式之类的东西。他们也可能缺乏更加概念化的领域知识，如设计思维和设计批判。同样，很多开发者缺乏软件架构、安全、数据规划和设计模式等方面的基本技能。所以，虽然我认为很多局限于某些开发语言、开发工具和开发技术的课程是在浪费时间，但是这些核心技能是非常有用的。不仅如此，随着这个行业日趋成熟，我们需要的人应该具有越来越专业的技能，比如在设计研究、可用性测试和交互设计等方面的技能。虽然这些技能可以在工作中学习，但是能够把这些技能应用到一

定高度的机构少之又少。因此，如果你参加专门的课程可能会把这些技术学得更深入一些。

Wolfgang Bartelme：一般说来，接受正规教育是件好事。然而，这只是开始，你仍然需要通过参加讨论会、读书，更重要的是，通过尝试新东西进行自我提高。

“那些不被所用工具限制的人，将会是最成功的人。”

Collis Ta’eed： 我不认为接受正规教育或培训是必需的。然而，这却是一种途径，无可厚非。就个人而言，我大部分靠自学，我仅仅参加过夜校课程，以便我在简历上有些可写的东西。但是最终，并没有人问过我的资历。实际上，人们一直通过我的工作表现来评价我。

对于那些正准备自学的设计者，我非常推荐去当地图书馆借一些关于设计基础的书籍。有很多需要学习的东西在网上是找不到的，尤其是关于设计历史和设计演化的知识。

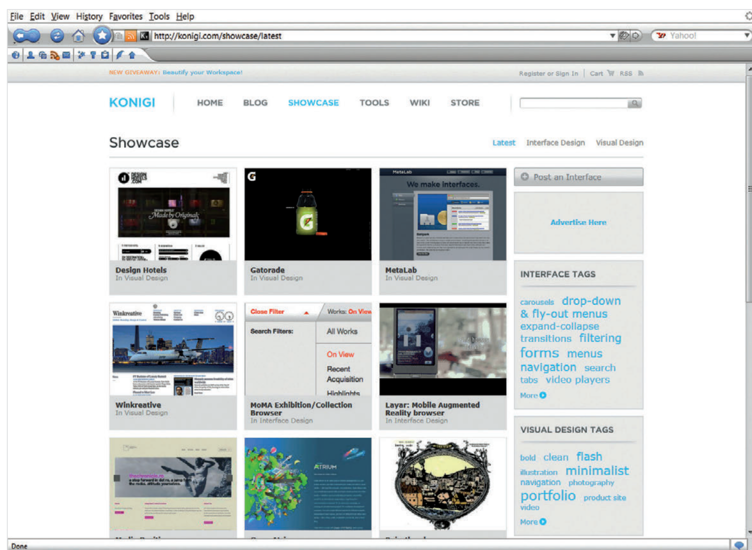
9.3.2　作为设计师你们是如何应对不断的改进和进化带来的挑战?

Liam McKay： 作为一个设计师，我努力提高的一个方式是对自己承诺，在每次设计中都尝试一些新的东西。在我开自己的公司 Function 之前，曾有一个阶段，我对自己的技术非常有信心，我做的设计都非常相似。在每个设计中，我使用相同的蓝色色调，使用相同的字体、效果和风格，这导致我不再能从设计中感受到乐趣，我知道我必须采取一点措施，所以我决定从此以后，在我所做的每个设计中，都要包含一些我以前从未用过的东西。尝试新东西的另一个好处是你的作品集会变得丰富多彩。不同的风格使你的作品增色不少，面对潜在客户，你会变得更有吸引力。

我想可以肯定地说，所有的设计者总是尽力完善他们的作品，在这一过程中也得到了成长。可能会产生障碍的主要是态度：在行业内，你永远不会找到高高在上的感觉，总是有新的东西需要学习和尝试。我非常清楚我还没有发挥到极致。并不是有什么事情使我不快，我总是会确立新的目标并为之努力。

Dan Rubin： 我总是在寻找那些与设计直接或间接相关的最新兴趣和创新尝试。我把自己当做一个一般意义上的“设计师”，而不是一个媒体方面的设

计师。在生活中，在我认为应该投入精力的方面有很多，我把自己看成是一个有创意的问题解决者。经常通过探索新事物进行自我挑战（比如摄影、探险和研究心理学），同时关于这些内容进行写作和教学，这使我作为一个设计师和作为一个人，能够进行自我提高和完善。另外，尝试不同的东西也会带来更多的乐趣。



Konigi 是诸多设计展示网站中的一个，此网站是关于用户界面设计的。

Andy Budd：我不认为变化会像人们想象的那样快。然而，如果你希望在 20 个不同的领域都保持领先，那就困难了。所以，努力做好某些事情并且专注于那些事情。这样，你将始终处于队伍的前列而不是一路追赶。博客很显然是一种与最新趋势保持同步的好资源，但是它的数量太大了，会使你无所适从。这就是为什么我建议人们把精力投入到专业的开发中去，并且要参加研讨会和讲习班。当然，这些你都可以通过资源丰富的互联网学到。如果花一些可以支配的时间就可以轻易掌握这些知识（并很快可以学以致用），何乐而不为呢？

Nathan Smith：老实讲，我尽量不去读太多设计博客中排名在前 10 位的文

章。虽然这些文章对于寻找关于设计软件诸多使用技巧的教程很有帮助，但是对于那些能提高技能的设计原理它们却很少涉及。我会到别处去寻找灵感。

网站设计还是相对年轻的行业。因此，我认为学习设计最好的方式就是以超越的眼光看待设计。是懒惰驱使我去 CSS 在线画廊寻找别人做过的东西并进行模仿。

有人说“艺术是用来欣赏的，设计是用来使用的。”然而，设计需要相当高的审美标准，我认为我们网站设计领域中很多人所做的设计是用来崇拜的。我们希望获奖，希望自己的作品出现在在线画廊的首页，并获得五星评价。我们希望人们在 Twitter 上反复讨论我们的教程。但实际上，对于终端用户而言，这些都无关紧要。伟大的设计是透明的。我仍然在学习如何才能不沉迷其中。借用 Joseph Ballay 关于玛雅设计的一句名言：“是设计的透明实现了它的功能”。

Keinth Robinson：我一直尝试新的东西并全神贯注地进行创造。对于一个设计师而言，客户项目和日常工作都可能成为真正的挑战。我每天会抽出时间进行总结。更好的情况是，当我接了一个设计项目时，我会竭尽所能地去做好。这需要自制力和努力，但是这些超出常规的项目会真正帮助一个设计师成长。

9.3.3　是什么把一个优秀的设计师和杰出的设计师区分开?

Dan Rubin：要理解设计不仅仅是视觉的，视觉设计和图形设计仅仅是我们作为设计师所做工作的一小部分。当人们要求我对“设计”进行定义时，我告诉他们的答案是：用有创意的方式解决问题。无论你是在设计产品、网站应用程序、出版物还是汽车，视觉效果都是方案执行后的结果。要知道作为设计师，你的第一角色是“问题解决者”，而不是“插画师”或“画东西的人”，或者甚至以为自己是“Photoshop 大师”，认识到了这一点，意味着你的定位是正确的。

Paul Boag：着迷、耗时、移情。杰出的设计师对细节着迷。他们会花若干小时进行微调，这些工作永远不会被察觉但是却会让整体的设计更富魅力。他们也会为设计添加能产生非凡效果的元素。这些装饰不会被每个人注意到，

但是注意到的人就会记住这个设计。

当然，对细节着迷是需要花时间的。我认为大多数客户不愿意为杰出的设计付费，这真是一种耻辱。有时候如果为了创造出确实令人惊叹的东西，额外花费的时间只能由设计师自己承担。

最后，我认为对于一个设计师而言，能够移情是很重要的。我们最终不是在为我们自己进行设计，也不是为客户进行设计。我们在为目标用户进行设计。真正理解这些用户以及能从他们的感受出发，对我们而言是至关重要的。优秀的设计师会做可用性测试，杰出的设计师也做，但是他能通过与用户进行情感交流来指导设计。



Freelanceswitch 的比率计算器。

Jay Hilgert：如果问我如何区分一个优秀的设计师和杰出的设计师，我的答案是：不断学习的渴望。在这一行，门槛总是在不停地提高，软件总在不停地进化。一个杰出的设计师总是热切希望学到更多的东西，并日复一日地不断提高。

Nick La：优秀的设计师跟随潮流。杰出的设计师创造潮流。

9.4 发展事业和做自由职业者

9.4.1 你们对那些开始单干或开始创业的设计师的建议是什么？该做什么和不该做什么？

Henry Jones：对于那些想要创业的人我有一些建议。第一，建立好的形象，或者说要在设计社区里具有一定的知名度。这也许看上去像一个难以完成的任务，但是如果你有一些合适的技巧，这其实很容易操作。有很多方法可以做到这一点，但是也许最容易的就是在知名的设计画廊中展示自己的作品或项目，这可以使你的作品每天被数以千计的人看到，其中既包括设计师也包括潜在的客户。实际上通过浏览这些网站以寻找设计天才的潜在客户的数量大得惊人。一旦你建立了一个较好的形象，你会发现新的项目会让你应接不暇。

第二，量力而为。一旦咨询服务的邮件开始往来，你会本能地对所有问题都说“是”，因为对于新兴企业而言，建立好的客户基础是很重要的。然而，这样做可能会损害你的名誉。如果你不能履行承诺或者你交付的项目不是每一部分都具有很高的质量。一种更好的方式是只接你感兴趣的项目和你力所能及的项目。

设计社区当然是一个很不错的地方，它总在寻找新的信息、资源、意见和灵感。

Liam McKay：当我自己单干、开博客和开一个小的设计公司时，我实际上并没有真正的计划。我有的只是信心和对优秀设计的深刻理解。我在原来工作的时候存了一些钱，决定花一些时间创建一个自己的博客，并在博客上发布一些免费的资源。我非常了解稳定的流量能够带来许多潜在的客户，所以我发布免费图标和免费 WordPress 主题背后的目的纯粹就是为了流量。参与很容易，无论你是通过创建自己的博客或网站还是通过参与其他人的博客。如果你要建立自己的博客或设计公司，忽略参与设计社区和开发社区可能带来的影响是不明智的。在与我的合作者 Spencer 合作建立了我们的博客后，我意识到他在 WordPress 方面是多么擅长（同时我发现 WordPress 作为一个 CMS 用途是那么广泛），我决定专注地做我们的长项——优秀的设计和对 WordPress 的深入开发。因为我们专注于这些领域，我们接到的很多单子也

都是基于 WordPress 的，这使报价、议价和与客户打交道变得更加容易。人们知道想要自己什么：他们知道 WordPress 是什么、它能干什么以及我们能为他们做什么。所以毫无疑问，把我们自己和我们最擅长的两件事结合起来是个好主意，因为这使我们能够专注于我们的长项，这也意味着可以使客户非常了解我们能为他们做什么。

“开办设计公司需要记住的一件事情是，这是一项事业。”

Jonathan Snook：开办设计公司最需要记住的一件事情是，这是一项事业。设计和接单并不困难，但是你还得能够管理现金流、纳税、合同转包以及其他的事情。你还需要确认你的收费和你的劳动是相当的。如果你过去全职工作的收入是每小时 25 美金，那你现在每小时收 35 美金才是划算的。不幸的是，当你开始审视自己的时间分配时，你会（也许有一点）意识到每天能够用来做工作的时间是多么少。每小时 35 美金的收费按一天的时间来平均，相当于每小时才 10 美金。

Chris Spooner：首先，对自己的技术能力有信心是提供给客户服务的好起点。在开工以后，牢记这一点也是个不错的主意，那些编程语言和设计风格你都不太熟悉的项目，最好不要去做。牢记这一点，对于描述你的专业技能和界定你的工作方向是很有帮助的。

与行业保持同步同样很重要。尤其是伴随着全球网站设计的飞速发展。对即将到来的变革、风格和新语言有所了解，将使你处于行业的领先地位。做一个积极的博主或社会化媒体的用户，这是一种自我充实、分享知识的好方式。

Andy Budd：自由职业市场已经极度饱和，要脱颖而出，你需要对你所做的东西非常精通，并且能够证明这一点。这意味着为重要客户创建一个作品集，并在杂志上、在线画廊上和活动中展示它。仅仅专注于目标客户很难高质量地完成这个任务，所以我推荐与那些能为你提供更大、更复杂的项目的机构合作。我建议设计者和开发者要专业化。比如，在你特定的领域树立目标，做最好的设计师、前端开发者或 Ruby 开发者，这样人们才会关注你。

“与行业保持同步是重要的，尤其是伴随着全球网站设计的飞速发展。”

Jon Hicks：我的最大的建议是，忽略财务将使你面临危险。确保你从每张发票中留出了足够的钱，用来支付税单，确保你的文案井然有序。如果你自己做账有拿不准的地方，不要在财务软件上浪费时间，去雇一个记账员或会计吧。如果你把他们的开销和你做这些事每小时的工资成本做一下对比，你会倾向于怎么做？他们还能够为你经营企业和节省税款提供建议。

9.4.2 在你们设计展示网站时，主要遵循的原则是什么？

Larissa Meek：展示你最好的作品，而不是展示你全部的作品。通过个案分析突出你设计选择背后的策略。

- 列出你使用的技术和软件。
- 如果你有不同的作品集，通过提供现场过滤来排列作品是很有帮助的（比如通过 Flash、CSS/HTML 和移动开发等条件进行过滤）。
- 如果是个人作品展示，列出你在每个项目中担任的角色。

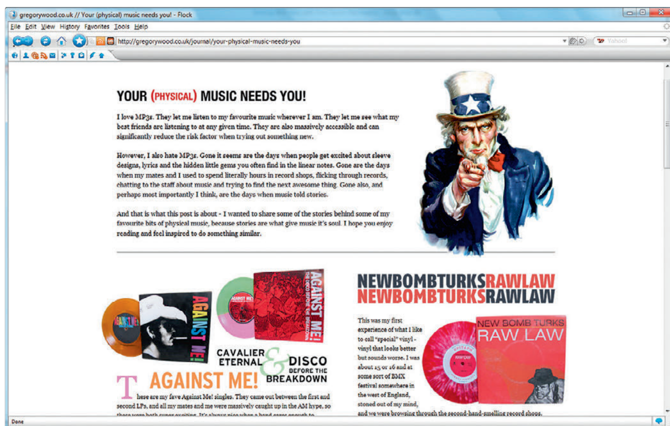
Elliot Jay Stocks：在提供图片的同时要提供链接（实际上，任何作品集只显示图片而没有提供可供检验的链接，在我看来是不好的，也是不太可信的）。让作品闪光，让内容呼吸。对于你在项目中做了或没做的事情要诚实。如果不是特殊的案例，用最少的文字为每个作品集添加一个说明。把案例研究控制在最小的数量之内——5 个就够了，如果多了就没有人愿意看了。

Veerle Pieters：首先，让你的工作成为你的代言人，因为它是你的名片。如果你是一名网站设计师，最好的做法是使用 W3C 标准。留意微小的细节，比如对版式的思考，这显示了对工作的热爱。不要使用完全基于 Flash 的网站，要恰到好处地使用 Flash。不要因为太过创新而使事情变得过于复杂。从客户的角度考虑问题，考虑一下要对你的作品有一个了解需要几个步骤。不要害怕用大的尺寸展示你的作品，这样潜在的客户才能真正看清细节，而不至于因图片太小而耗费眼力。对于你在工作中的角色要诚实。太多时候，我看到这样的例子，设计师对他们在作品中的贡献含糊其辞，以至于客户认为他们完成了全部的工作。从长远来看这帮不了你。分工没有什么不对，你没有必要靠谎称自己做了所有的工作获得成功。



来自美国科罗拉多的 Shannon Moeller 的作品集。

Chris Spooner： 因为展示网站是对你作为设计师的技能和知识的展示，我认为它的主要原则是通过设计展示网站本身的设计实践，展示你的设计理论知识和对设计的理解。然而，我也认为整个作品集的设计不应该过于显眼。所以，我想在布局和版式方面，恰到好处的轻量级设计是一种不错选择。一页清晰易懂的“关于”页面（About），对任何作品集而言也是有用的，这是把你自己介绍给潜在客户的一个好办法，它使你和用户可以进行互动。



Gregory Wood 的博客通过非常规的布局给访客留下了深刻的印象，每一篇博客都有自己独特的版式。

幕后故事

第10章

Smashing Magazine 是如何诞生的？我们是怎样工作的以及幕后都发生了哪些事情？我们成功的秘诀是什么？我们一直在回答读者提出的这些问题。

Smashing Magazine 的故事不是那种经典的类型——两个小伙子在合适的时间、合适的地点想到了一个非常好的主意。这是一个关于专注、耐心和努力工作——尤其是努力工作的故事。在 Vitaly 最近参加的一次会议上，一位充满活力的年轻人问他 Smashing Magazine 的成功秘诀是什么。坦诚起见，Vitaly 竭尽所能地解释说，没有什么秘诀可言，只不过是充满激情并专注地投入到你所做的工作中去。Vitaly 指出了努力工作的重要性，最重要的是永远不要放弃，如果你确认正在做正确的事情。接下来他又问了一些问题，试图获得更多具体的、可量化的见解，突然这个年轻人走开了，消失在人群中，也许他觉得他的问题没有被认真对待。

事实上，Smashing Magazine 没有什么特别的成功秘诀。我们有明确的目标，并且清楚地知道该如何实现这些目标，但是整个过程中最重要的品质是，实现“设计社区”预期的强烈愿望，而不管这会耗费多少时间和金钱。这个背后没有什么高明的、秘而不宣的策略，也没有幕后操盘手来决定哪篇文章最适合这个或那个市场活动。Smashing Magazine 现在和未来将一直是独立的。每篇文章的质量完全由其内容的实用度来衡量。这是我们工作的主要原则。我们努力工作，以确保我们在这一原则问题上永不妥协。

10.1　一切是如何开始的

关于 Smashing Magazine 的诞生，最不寻常的事也许是我们从来没有坐下来一块讨论过整个事情。我们从来没有竖起白板，对一份有着明确市场模式的成功杂志进行头脑风暴。实际上，Smashing Magazine 是一次随机试验的结果，它由两个志同道合的网站工作者发起，他们具有共同的激情和知识，丰富的经验（Sven）和饱满的干劲（Vitaly）。

一切都开始于 2006 年 8 月。Vitaly Friedman，一个计算机专业的学生和一个具有 8 年工作经验的自由网站设计师，偶尔开始为声名远播的德国在线杂志“Dr. Web ”撰写文章。这个杂志已经在讲德语的网站设计圈内驰名 12 余年。站在该杂志幕后的是自由设计师 Sven Lennarz，一个把编辑并且维护 Dr. Web[1] 当做自己全职工作的人。

他们两个都有各自的事务要处理，Vitaly 在萨尔布吕肯（在德国的西南部）做自由职业者，并且要通过大学考试。Sven 在吕贝克（在德国北部）从事网站设计开发并与作者讨论文章的构思以及实现 Dr. Web 的创意。

“每篇文章的质量完全由其内容实用度来衡量。”

此时，我们已经认识两年多，但是从来没有打过电话或见过面。我们偶尔借助电子邮件[2]，对最近的开发情况和新帖内容的构思进行讨论。因此，一个大的合作计划通过无数邮件中的一封被审慎地提出来并不足为奇。老实说，这个对我们两个来说都算不上什么大事。实际上，为设计者和开发者创办一份英语在线博客的想法是 Sven 提出的。当 Vitaly 建议用“Smashing”（“We smash you with the information that makes your life easier. ”）这个名字时[3]，我们都同意建一个简单的博客、发一些文章看看情况，以此来尝试一下这个想法。那时我们既没有经济上的支持，也没有可签的单子。

项目的大小细节，包括后端、前端以及针对发布内容的意见都是通过大量的电子邮件来讨论，同时也使用语音留言和手写便签，这也是为什么大多数邮件的标题都被截断的原因 (如“Re: Re: Re [34]”)，这也是为什么我们会需要这么多的邮件来达成一致的原因。非常有意思的是，新计划中各个模块的整合是由 Sven 和 Vitaly 在德国的小酒馆、比萨店、街道以及大学的课堂上（或课程之间）各自完成的。

两周的准备后，域名注册下来了，服务器配置好了，WordPress 也装上了。

[1] http://drweb.de。

[2] 现在的情形也没什么太大变化。大部分时间我们通过E-mail或偶尔使用Skype讨论杂志的文章和发展。事关重大变革的讨论通常在德国不同的城市面对面地进行。

[3] 译者注 ：这句话的含义是，“我们会用使你惊呆的信息简化你的生活”，smash一词在此处的含义是“使（某人）过于吃惊或高兴而说不出话来”。

我们采用了最先找到的 WordPress 最小化主题之一，它即刻就可以使用。Smashing Magazine 的网站在 2006 年 9 月 8 日上线。杂志当时甚至连 Logo 都没有：我们使用了 WordPress 模板默认的 Verdana 字体作为标题。最初的文章来自 Dr. Web，我们把德语翻译成英语，并把它们发布在新诞生的 Smashing Magazine 上。

这些最初发布的文章没有获得巨大的成功，当然，我们也没有指望它们能获得成功。我们没有公布或酝酿任何宏伟蓝图。我们继续处理各自的事务，保持耐心和决心。实际上，数月后我们才获得重大的突破。

10.2　我们是如何走到今天的

事情进展得很慢，毕竟，我们各自都有各自的事情要处理。最初，我们达成一致，每周最多发布两篇文章，以 Dr. Web 的文章为基础并补充新的内容。比如，我们扩充这些德语文章的内容并为国际读者润色它们。流量在最初的几个月缓慢地增长，但是直到 2006 年年底，我们的文章（那些早期发布的文章）开始出现在威力十足的 Digg 的首页之前，我们都没有看到流量的激增[1]。

我们第一次扬眉吐气是由于“2006 年 50 个漂亮的基于 CSS 的网站设计”（50 Beautiful CSS-Based Web Designs in 2006）这篇文章带来的巨大的访问量，该文出现在了 Digg 的首页，并获得了 3100 个投票。当然，由于 Digg 的效应，我们的服务器崩溃了几分钟。毕竟每一张图片都超过 150KB，提供服务的服务器没有为这样的流量高峰进行过精心配置。几周后，“53 种你不可或缺的 CSS 技术”（53 CSS Techniques You Couldn’t Live Without）的 5300 票的成绩简直使我们惊呆，这是 Smashing Magazine 发展中的转折点。从那以后，我们的文章在各个媒体上表现得很出色，它们带来流量，并通过广告位带来收入。在那个时候，我们知道，我们必须重新考虑曾为杂志制定的那些保守的计划。我们也知道，我们最终必须得碰一次面，这也是第一次碰面。

[1] 是的，你可能会为过去几年那些遍布在互联网上与设计相关的文章列表责怪我们。不是我们发明了这种格式，但是我们使它达到一个更为复杂的、可用的模式。

通过 Smashing Magazine 谋生并不是我们的想法。没有成功的压力，我们有足够的耐心，并能使杂志沿着正确的方向稳步前进。对于网站设计的经验和激情有助于我们不犯错误，并使我们坚持做各自的事情。当意识到设计社区对我们文章感兴趣时，我们向前迈进了一步，我们扩充了文章内容并优化了文章结构，包含了更多的图片和插图。一些读者抱怨（目前仍在抱怨）我们文章的长度使他们的 RSS 源阅读器和他们的电脑变慢。但这并未吓住我们，我们继续通过长篇的文章来使读者惊呆。



2007 年以前 Smashing Magazine Logo 的草图。

有些文章是如此之丰富，以至于其中的一些可以作为独立的网站，这也是为什么为了打印方便，我们开始以 PDF 的格式提供文章的原因。

当然，较长的文章意味着这个小项目需要更多的工作和更大的投入。我们在网上到处搜索墙纸、工具、CSS 和 JavaScript 技巧、教程和其他的资源。我们追溯并介绍设计趋势以及共同的可用性问题。此外，我们也寻找高级的题目，比如数据可视化，同时通过访谈设计师和专家获取更多的专业意见。至此，我们已经开发了一系列工具用来自动收集有用资源和准备设计展示。我们现在每周可以发布更多的文章。星期二和星期四专门用来发布巨型的文章，我们决定通过每周的“星期一启示”系列为读者提供图片饱满的启发性内容。



Smashing Magazine 文章的全部插图。这里展示的是一个集合中的 650 张图片。

当然，更多的图片意味着更大的服务器负载。为了跟上访客不断提高的胃口，我们不得不经常性地扩充服务器。读者通常通过其他媒体网站发现我们：如果没有 Digg、StumbleUpon 和 Delicious，我们肯定无法获得这么多的流量。我们的文章因为实用或具有启发性，经常获得标记或推荐，这使我们可以借助各个媒体的力量传播内容并带来流量。Frederick Townes、Neil Patel、Muhammad Saleem、Victor Battera、Dave (BlueNile)、Tal Siach 和其他一些朋友帮助我们更好地理解社会化媒体，以及这些媒体在互联网上的作用。

2007 年年中，我们终于不再使用 Dr. Web 上的文章了。此前，我们已经发布的文章是翻译和原创的混合体：Vitaly 重新写文章，Sven 用德语写文章，然后 Vitaly 把它们翻译成英语。Smashing Magazine 的读者很快习惯了我们标志性的巨型文章。很多 Dr. Web 的文章满足不了这种新的样式：它们太短，

而且其中一些讨论的更多的是项目管理、搜索引擎优化（SEO）和网站开发市场，而不是网站设计。所以，我们渐渐转向为 Smashing Magazine 定制原创内容。

我们很快发现，对英语的掌握程度不像想象的那样好。越来越多的读者开始抱怨语法错误、拼写错误和含糊不清的句子。所以，我们开始寻找一位专业的校对者。不幸的是，我们用了一年的时间才找到一个合适的人选，Andrew Lobo，从 2008 年 11 月开始，他出色地编辑了 Smashing Magazine 的大部分文章。

根据用户的期望，我们发布内容的频率提高了。2007 年年底，我们意识到读者数量已经非常可观，形成了一个极为多样化的读者群。通过每周发布的三两篇文章来满足这些读者的预期开始变得困难。从那时起，我们开始聘用固定的作者。

Sean Hodge 和 Mark Bloomfield 是最早的访客作者，他们帮我们开阔了 Smashing Magazine 的视野、主题和格式。在他们的帮助下，我们每周能发布 4、5 篇文章。一旦这些访客的文章被发布，我们就开始收到来自全球设计者和开发者的需求。因为自由写作通常是一项间断性的工作，很多设计者可以在项目之间完成。当新的项目开始时，作者如果没有时间就可以停下来。

因此，我们总是在寻找新的作者，这并不是一件容易的事。为了寻找能够经常为我们写作的作者，我们花了很长时间。我们既被动地寻找（回答电子邮件中的疑问），也主动地寻找（搜寻对设计、可用性和代码有深入了解的博客作者）。我们甚至举办了一次 Smashing 作者大赛。比赛中，我们的读者可以以访客身份提交内容。我们发布那些最有意思的内容，最佳文章的作者（由读者评选出）将获得一台 Apple MacBook Air。奖品（以及它的运送）本身就是一个故事，因为以“10 个顶级的 CSS 表格设计”（Top 10 CSS Table Designs）获胜的 R. Christie 居住在印度尼西亚。

比较讽刺的是，我们确实没有通过比赛本身找到稳定的作者，但是我们吸引了全球许多自由作家的关注。数月后，我们聘用了一批优秀的稳定作者，比如 Steven Snell 和 Jacob Gube。后来，其他的作者也开始为我们写作(Vailancio Rodrigues、Jean-Baptiste Jung、Chris Coyier、Kayla Knight、Inayaili de Leon、

Noura Yehia、Aquil Akhter、Danny Outlaw、Cameron Chapman、Glen Stansberry 和 Matt Cronin）。我们主动地搜寻为我们带来了 Chris Spooner 和 Dmitry Fadeyev，他们也是本书的作者。这些新的作者不仅为 Smashing Magazine 注入了创意，同时也为它带来了新生，使它能覆盖我们两人都不太了解的领域，并且为我们腾出时间，使我们能够就那些比较贴近我们内心的主题写一些文章。

但是由于包含了我们所不熟悉的主题，维护高质量的内容变成了一种挑战，因为我们的确不知道这些内容是否正确、是否体现了最佳的设计形式和代码实现。所以，在 2008 年年底，我们开始邀请专业的设计师和代码开发人员，对一些文章的内容进行咨询和编辑。例如，PHP 大师 Chris Shiflett 和 Sean Coates 发现一篇关于 PHP 的文章中有一些严重的错误，我们邀请他们写了一篇文章更正。当文章发布时，我们利用这个机会为我们的失误和由此产生的不便公开道歉。从那以后，我们相当谨慎地处理不熟悉的题目，在发布之前尽可能地向专家咨询意见。



Smashing Magazine 两周年的海报，由 James White 于 2008 年 8 月设计。

截止到 2008 年，我们的发布计划已经提升到每周 5 ~ 10 篇文章（过去周末是不发布文章的，然而，甚至这一点最后也改变了）。从那以后，我们定期收到应征作者职位的信件，我们从来没有错过任何一个可以尝试并看看效果如何的机会。在 2007 年到 2009 年之间，我们发表的文章出自遍布世界各大洲的 120 位作者之手。遗憾的是，我们从来没有见过他们中的任何一位，我们希望这种情况在将来能够有所改观。

当然，除了 Smashing 作者大赛，我们还组织了其他的比赛，来给我们的读者一个亲自参与以及创作绝佳内容的机会。在与读者的共同努力下，我们已经发布了页首图形、纹理、字体模板和 <hr /> 标题作为自由下载资源。正如你预期的那样，这些竞赛的获胜者将会获得非常酷的奖品。也许我们举行的最著名的比赛是设计桌面日历墙纸，每个月来自全球的设计者都会把他们的作品发给我们，我们会在下个月之前发布这些作品。这个主意是 Vitaly 向 Sven 提议的，起初 Sven 对此表示怀疑。实际上，我们计划活动只持续 12 个月，但是当我们的读者希望继续时，我们也就无法停止了。有时候我们仅仅是为



桌面日历壁纸是 Smashing Magazine 最成功的系列之一（注意这两张图片需要合并一下）。在 Smashing Magazine 上最成功的一些作品。

了发奖品而发奖品。多年来，通过将奖品赠给杰出的读者，使我们感受到了极大的乐趣。许多年前，我们从来没有设想过我们此后会给出这么多高质量的赠品。一些赠品并非出自我们之手，而是来自设计者的捐赠。作为回报，我们为他们带来巨大的流量和极大知名度。实际上，这是一笔相当好的交易，特别是赠品设计得还不错的时候，将会最大程度上（在设计者和发布者之间）产生双赢。Elena Gafita 的 WordPress 主题和 Jos Buivenga 的免费字体是我们最早发布的赠品。从那时开始，我们已经发布了 130 多种高质量的赠品，它们来自全球 100 多位设计师，我们诚挚地感谢他们为之付出的辛苦劳动，同时感谢他们允许社区用户可以免费获取这些资源。

当然，在这个过程中我们也遇到了困难。多年来，Smashing Magazine 收到了 44000 多封邮件，回复了 17000 封邮件（平均每天收到 40 封，回复 15 封）。因为只有我们两个人处理邮件，所以我们不得不按优先级处理邮件。尽管我们尽力立即回复最重要的邮件（否则它们将排成长队以引起我们的注意），但是仍然有许多邮件没有得到及时回复（或根本就没回复）。我们试图仔细查阅每一封我们收到的邮件，但是有的时候这是不可能的。其间，我们经历了三次严重的数据库崩溃（谢天谢地，我们有备份），以及由于两块硬盘驱动器过热而导致的数据丢失。因为这个原因，一些即将完成的文章再也无法发布。非常不幸，它们永远无法触及 WordPress 的引擎了。

并不是所有人都知道像 Smashing Magazine 这样的项目不能发布到一般的网站空间或者托管服务器上。

我们遇到的另一个问题是杂志不断增长的流量。在 2007 年 12 月，由于两次 Digg 效应、一次 Reddit 效应，一次 StumbleUpon 效应以及一次巨大的 Slash-dot 效应，所有的加在一起，我们可爱的服务提供商对我们的服务器进行了一段时间的限流，并最终把我们赶了出来。很显然，由于负载过大服务器挂机了。我们从来没有查出到底发生了什么或我们为什么被关机。很显然，我们小小的博客导致了服务器性能（在当时）出现问题，它吸引了太多的流量（那时还是支付固定额度的流量费），这对运营商来说并不划算。我们需要的更多、更多、更多。很显然支持 TB 级流量的费用是高昂的。

一段时间里，Google 也不喜欢我们。我们在没有使用“no follow”属性标签的情况下销售文字链接，这导致了在 2007 年网站被 Google 搜索引擎屏蔽。

我们很快移除了这些文字链接，几个月后我们又出现在了搜索引擎里面。

目前，我们的月流量在 TB 级能保持在两位数的左右，我们有定制的服务器，主站放在德国，离 Vitaly 住处不远的地方。我们有 6 个服务器不间断地向读者传送图片和内容。这些服务器的配置、维护和优化由 Rene Schmidt 完成，他也是本书的作者之一。他还负责定制我们主服务器上的 WordPress 插件，这些插件为每周数以百万的页面请求服务。

当我们写这一章时，Smashing Magazine 中一共包含了 761 篇文章，附带一些页面。这并不多，但是这 761 篇文章带来了 130000 条评论和转发。很多评论产生于 2009 年 2 月，在我们赠送礼品期间：我们希望关于那个帖子有尽可能多的评论，最终我们如愿以偿。当评论达到 8999 条时，WordPress 崩溃了。

10.3　我们成功故事的关键因素

虽然一个新的想法有很多途径变成一个成功的故事，有一件事情是明确的：成功需要大量艰辛的工作（这不算什么重大发现，但还是值得提出来）。虽然，我们的一些读者对于成功另有高见，但是我们总是避免。有时，如果能够获取独有的、有价值的内容，我们故意选择比较困难的方式（比如，当我们准备进行调查和个案研究时，每次都要耗费近 60 个小时的时间，我们发现这是一个很难打破的纪录）。

对我们和读者而言，Smashing Magazine 外观不像发布的内容那样重要。

实际上，有价值的内容——或者更精确地说，写作、编辑和发布有价值的内容——是我们多年来工作的重心。内容当然是成功的关键，在我们的案例中，它就是真正的终极要素。事实上，Smashing Magazine 从来没有过摄人心魄的或者十分美观的设计：我们用的是滑稽的大标题，我们的搜索引擎多年来表现平平，一些设计元素并未连贯地对齐而且风格不一致。

但是所有的这些都不要紧，对我们和读者而言，Smashing Magazine 的外观不像其发布的内容那样重要。在完全忽略视觉设计的同时，我们把更多的精力投在稳定的可用性和功能性上。最重要的是，有价值的、中肯的内容，是

驱动访客一而再、再而三地回来访问我们的网站。

我们初期得到的一个教训是，要对来自各方诱惑性的提议说“不”。当你刚刚走上正轨，你就立刻会被一大群想分享你的成功的鲨鱼包围，他们会通过非常有说服力的条件引起你的注意。实际上，除了分享你的成功和能为他们带来流量或收入的所谓的“伙伴关系”之外，这些鲨鱼甚至不知道他们想从你那里得到什么。从一开始我们决定始终如一地做自己的事情，并且把它做好。预先避免分心有助于我们在将来避免问题。

如果我们必须定义这些年来对我们工作影响最大的原则，毫无疑问它将是我们不遗余力地去理解客户并实现他们的预期。我们坚持不懈地倾听读者关于 Smashing Magazine 的提问和抱怨，我们甚至监视发生在杂志之外的话题。比如发生在博客、论坛和社会化媒体中的话题。我们坚持不懈地对所触及的信息进行分析和分级，进一步处理那些可能有用的信息，并抛弃那些对我们的读者无益的信息。显然，当大部分（尤其是多样化）读者的已经形成了很高的期望时，使他们失望变得非常容易。读者似乎总是记住那些评论截然相反的、有争议的文章。这些文章的被评论的次数总是高居榜首，经常包含类似“史上最差”之类的字眼。



我们著名的 RSS 源图标，由来自巴西的 Dirceu Veiga 设计，由 Smashing Magazine 于 2008 年免费发布。

所以，要找到正确的发展方向，理解读者的兴趣和需求变得特别重要。要获得这种洞察力，最简单的方式之一便是把自己放到读者的位置[1]。在我们的案例中，这很容易做到。我们自己就是网站开发者，依据我们的专业知识做出判断，我们决定专注于那些具有最高实用价值的内容。事情的结果是全球数以千计的设计者和我们着有相似的兴趣，我们的专业知识很显然对他们很有用。从一开始，实用性标准就是我们工作不可违背的前提，它已经确定了我们所发布内容的本质。

我们的读者经常会问我们是如何与他们心有灵犀的：也就是，他们在搜寻某一方面时，我们恰好发布一篇有深度的文章。我们的目标是传递实用而且可用的内容，所以我们发布的文章正好是，一些设计者和开发者正在搜寻的资源并不足为奇。我们了解读者，读者也发现有份杂志为他们的日常工作提供了帮助。最后，这个不难实现，因为我们曾经对读者的需求有过真诚的探求（现在也是如此）。

因为这个原因，在 2008 年 8 月，我们决定尝试 Twitter（这实际上已经很晚了）。我们开始了解各种个人信息聚合（life-streaming）程序，并想试用一个看看效果如何。坦白地说，我们一开始持怀疑的态度但最终习惯了它。我们拿这种服务做实验，努力探索如何使用它更好地跟读者交流。最后，我们想出了很多主意，目的在于把读者更加紧密地整合到 Smashing Magazine 的日常事务中。

我们现在在 Twitter 上分享有用的点子、幕后的创意和个人见地。

我们要求 Twitter 的粉丝积极参与决策过程。比如，我们的粉丝可以决定接下来发布哪篇文章，并对未来要发布的内容提供有趣的建议。我们也在杂志的“Ask SM”部分对读者的难题和问题进行跟进。Twitter 已经变成了我们和读者交流的主要渠道，并使我们能够更加了解读者，使读者参与进来，以及使我们更好地预见他们的需求和兴趣。

[1] 当然，理解读者绝非是放之四海而皆准的黄金法则。比如，许多艺术家采用更加自我专注的方式，仅仅做他们想做的事情，完全忽略公众的意见。在我们的案例中，理解设计师的需求是至关重要的，否则，我们将根本无法达到观众的需求。

10.4 幕后在发生着什么?

在 Smashing Magazine，质量控制始终是一件至关重要的事情，几乎占据了我们工作一半以上的时间。我们把引导读者的兴趣和确保发布的文章正确、可用、清晰作为我们的任务。最后一点解释了为什么每次为了达到真正使人惊呆的效果，大多数的发布内容要花费 20 个小时以上的时间以去准备、修订、编辑、更新以及一遍遍地重新编辑。

需要这么长的时间去准备内容的另一原因是，这些内容经常是由遍布全球的一大群作者合作完成的。我们经常和来自欧洲、大洋洲、亚洲和南美洲的撰稿人合作，以确保内容的丰富多彩和与众不同。我们也尝试把 Twitter 粉丝的反馈也整合进来。编辑团队仔细审阅每份提交上来的内容，如果反馈内容可接受，原来的文章将被改写并追加上新内容。

好的东西往往还可以更好，我们的作者同样也会经常注意到数不清的、由编辑团队提议的改写、扩充和（但愿是）改进，即使是在发布文章之后，即便文章可能已经更新了 5 个版本。我们每天计划发布的文章数量是不固定的，我们并不在乎文章的作者多有名望：我们的目标是传递高质量的内容。

不幸的是，我们的判断有时会误导我们，我们也会犯错，比如问作者错误的问题或是为读者提供错误的答案。但是我们知道我们会对发布的每一个字或每一张图片负责。

我们非常欢迎任何建设性的意见，以及所有那些能使我们变得更加明智、更强或在未来避免同类错误的反馈。出于对质量的考虑，我们也会不断检查我们的准则，并在任何必要的时侯对它进行完善和订正。我们从自己的错误中学习，并能足够坦诚地承认错误，并为错误道歉，而且在以后会尽力改正，这是我们哲学中重要的一部分。

这看上去可能像一个严格的、高难度的工作流程，但实际上它营造了一个非常具有创意的、多产的环境。根据我们的经验，最初跟作者的沟通，对于获得一篇好的文章来说是至关重要的，所以我们经常鼓励作者提问并与我们分享他们的难题。我们也经常向作者推荐有趣的观点和方向，也许这是他们并没想到的。

smashing magazine

we smash you with the information which will make your life easier. really.

(X)HTML template systems in PHP »

// october 1st, 2006

Web developers find themselves sooner or later in a dire situation: their scripts become an entangled mess of code. Program logic, presentation logic and in a worst case scenario even layout and design are so interconnected with each other that the further development becomes difficult. Would one have thought ahead and had created a clear structure for the code - the so-called HTML-template-engines, the integrity of the layers would have been preserved.

A HTML-template-engine serves to separate program logic (PHP), from layout, (HTML) and design (CSS). Most existing template-Systems are not limited to HTML-models. They are generally usable for E-Mail and even XML.

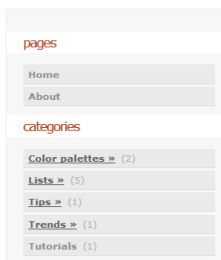
Program code can be changed independently of the web site design. All this is theory though. In practice and complex projects this division can not be realized most of the time. To prevent inflating the program code unnecessarily with representation logic, another fourth level is often added. In this layer logical connections are processed, which directly affect the representation. Example: Table line paging or dynamic lists.

A further substantial advantage of HTML templates is the reusability of program code, design and layout. A template can be used in numerous places on a web site. Changes in a template affect all pages where this template is being used. This can save a considerable amount of time and money, because instead of several pages, just the template has to be changed and proofed.

Levels

Program logic
On this level typical script tasks are settled, for example retrieval and conversion of data from the data-base. If the HTML template system is initiated, variable html templates are assigned to the appropriate substitute symbols.

Layout-level
It consists of models, for example of HTML. The model of a specific site can be composed of several templates, which are joined in program logic more directly by the template system. The layout too will partly be determined on this level;





News-Feed

Previous Posts

- HTML Template Systems
- List of Best SEO-Tools
- Color Palettes: Insurance
- The List Of Lists
- Color Palettes: Traveling
- Symbols, Buttons & Icons
- Web 2.0 Design Trends
- List of CSS Tools
- Nifty Tools & Diagrams
- 1 User Online

早期的 Smashing Magazine。这就是追溯到 2006 年 11 月时我们为它设计的样子。

我们的供稿者对所写的题目感到得心应手，并为所写的内容感到兴奋，这一点对我们来说很重要。为了避免写作过程中产生误解，我们为作者提供了一系列的写作模板和一份写作风格指南，它们囊括了我们的要求和预期。

相应的，我们没有通过设置截稿时间来限制作者的创造力。而由作者决定写一篇文章需要多长时间以及需要投入多少努力。我们非常清楚该如何对巨大的努力和奉献进行答谢，这也是为什么那些耗时的高质量文章会得到慷慨的奖金。我们工作的一条主要原则就是公正地对待我们的作者。在对作者的出色工作进行奖励的时候我们从不犹豫。比如，在他们为 Smashing Mazagine 完成第 10 篇文章时付给他们一笔可观的奖金。我们在作者身上花了大量的金钱，他们对每一分钱都受之无愧。

我们也关注那些读者感兴趣的、高质量的内容。那些经久不衰让人受益匪浅的文章和资源会受到本地或全球读者的关注和评估（一些内容要经过数月的准备，新的工具或技巧会一天天地被添加进去）。了解题目之间的联系、决定文章应该采用的最佳格式、把这些写下来并传达给作者或者自己来写文章，这些都需要花费时间。比如，当浏览网站设计（直投给我们或我们在线浏览）时，我们会关注最小的细节，如评论表单的设计、按钮的摆放、字体、布局、设计元素的对齐。我们会把所有这些都保存在一个原创创意库之中，这为产生新内容奠定基础。

10.5　感谢你，设计社区

在 2008 年一年中，我们收到了一封长信，是一位来自南非的平面设计教师写的，他在信中抱怨了他所在地区教学资源的不足。他告诉我们很多个月以来，每天晚上他都会从 Smashing Magazine 挑选一篇文章，并为和学生进行详细讨论而做好准备。这是他传授优秀平面设计主要方式之一。除了一些图书馆的旧书外，Smashing Magazine 是他课程的唯一参考资源。他写信的原因很简单，就是表达他和他的学生的诚挚谢意，感谢设计社区的巨大支持、大量可免费获取的文章、非同寻常的专注以及乐于助人的风范。这封令人感动的信引发了我们的思考：设计社区并不总是这样。随着设计博客和设计杂志的兴起，在今天作为一个设计师你永远都不会孤独，总有人会帮助你。作为设计师、开发者、发布者和读者，我们都应该认识到设计社区在它的发展过程中已经经历了一个令人瞩目的里程碑事件。也许我们还没有注意到这一点，我们已经做了一件了不起的工作——把我们自己联合起来、分享激情并传播最佳的设计实践知识以及有用的工具和资源。

也许，设计社区最难以置信、也最容易被忽略的品质是它有积极的、充满热情的精神。现在，每天数以千计的有天分的、辛勤工作的人，在那里通过他们的工作获取新的理解、实现非凡的创意并与其他设计师分享经验。正因为有了支持者和人际网络在可及范围内的帮助，设计社区已经发布了大量高质量的文章、资源和工具，并对任何人开放。每一个个体的奉献在支持社区，社区则通过流量和口碑营销——网络所具有的最大影响——来回馈这些奉献者。

事实上，离开这些奉献，Smashing Magazine 将无法存在。杂志最重要的元素永远是我们的读者。离开作者、数百位贡献者和数百万自网站上线以来就浏览网站并推荐杂志的访客的强烈支持和投入，杂志不可能走到今天。我们的工作不仅仅是为社区做贡献，我们还要帮助维护社区那能够使创意产生、创见交换、讨论发生的创造力十足的环境，并使设计者和开发者生活得更加惬意和富足。是你们开启它并使它成为可能。每一个贡献，即使是最小的，也会让未来大不相同。我们认识到了这一点，我们很尊重你们，并感谢你们的支持。在未来，我们将继续竭尽所能地坚持这一点。

设计原理和实践并重，交互设计师、视觉设计师、前端工程师和产品经理的必读之书！

余果，腾讯社交用户体验设计部 前端工程师

众妙之门

网站UI设计之道

（修订版）

知名设计网站Smashing Magazine上线以来的精华总结，来自全球的知名设计师多年积累宝贵经验的无私分享。

《众妙之门——网站UI设计之道》（修订版）共分10章，前8章讨论了和网站UI设计相关的8个不同的领域，覆盖面非常广，具有很强的操作性和专业性。第9章以访谈的形式介绍了多位专家对一系列重要问题的看法，入木三分，很有启发性。第10章以讲故事的方式，把Smashing Magazine的创办和发展过程娓娓道来，同时毫无保留地分享了两位创始人的成功秘诀和价值观，让人受益匪浅。全书逻辑严密、言简意赅，设计人员可以快速地找到自己想要的东西。

本书最大的价值在于它在传授具体经验的同时，也对设计思想和设计理念进行了深入的探讨，在“授人以鱼”的同时，不忘“授人以渔”。相信读完本书，读者一定会有一种相识恨晚的感觉。

封面设计：周亚宁



分类建议：计算机 / 网页设计
人民邮电出版社网址：www.ptpress.com.cn



欢迎加入

图灵社区

最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版的梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要你有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

最直接的读者交流平台

在图灵社区，你可以十分方便地写作文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、审读、评选等多种活动，赢取积分和银子，积累个人声望。

ituring.com.cn